Universidad de Cienfuegos “Carlos Rafael Rodríguez”

Facultad de Informática

Carrera de Ingeniería Informática

TÍTULO: TÉCNICA SEMI-SUPERVISADA PARA LA DETECCIÓN DE GENES ORTÓLOGOS.

Trabajo de diploma para optar por el título de Ingeniería en Informática

Autor(a):

Antoinette M. Prieto Aragones.

Tutor (a):

Ing. Annelys Gato Saura.

Consultante:

Dr. Eduardo Concepción Morales.

Cienfuegos, Cuba

Curso 2010 – 2011

Declaración de autoría

Yo Antoinette Prieto Aragones, me declaro como único autor del trabajo de diploma titulado “Técnica semi-supervisada para la detección de genes ortólogos”, y autorizo al Departamento de Informática de la Facultad de Informática en la Universidad de Cienfuegos “Carlos Rafael Rodríguez” para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los 6 días del mes de Julio del año 2011.

Antoinette Madonna Prieto Aragonés.

Nombre completo del autor

Ing. Annelys Gato Saura.

Nombre completo del tutor

Los abajo firmantes certificamos que el presente trabajo ha sido revisado según acuerdo de la dirección de nuestro centro y el mismo cumple los requisitos que debe tener un trabajo de esta envergadura referente a la temática señalada.

_____

Firma Tutor

_____ _____

Firma ICT Firma Vicedecano(a)

Pensamiento

"La mera formulación de un problema es la mayoría de las veces más importante que su solución, que puede ser simplemente una cuestión de habilidad matemática o experimental. Formular nuevas preguntas, nuevas posibilidades, mirar problemas antiguos desde un nuevo ángulo, requiere una imaginación creativa y marca verdaderos avances en la ciencia."

Albert Einstein

Dedicatoria

Dedico esta tesis especialmente a mis padres, por su amor incondicional, por todos los sacrificios que realizaron para educarme sin importarle las consecuencias, y por enseñarme, mediante el ejemplo, a no rendirme jamás. Es mucha la paciencia que han tenido para ver llegar este momento, pero finalmente aquí está, con todo

Mi corazón.

Agradecimientos

A mi mamá y a mi papá por todo lo que me han enseñado y por sus innumerables consejos.

A mi padrastro por ser tan bueno conmigo y soportar todas mis majaderías.

A mi novio Sergio Luis por brindarme su ayuda incondicional en los momentos más difíciles, por toda la paciencia que me ha tenido y por todo su amor.

A mi tía Raiza pues siempre ha estado ahí cuando la necesito.

A mi tutora Annelys por ayudarme tanto y tenerme tanta paciencia.

Al profesor Eduardo que sin su ayuda este trabajo no hubiera sido posible.

A mis compañeros de grupo, especialmente a Ileana, Liset y Yesica.

Doy gracias también a todas aquellas personas que de una u otra forma colaboraron conmigo y me ayudaron a concluir exitosamente este trabajo.

Resumen

El presente trabajo tiene como título “Técnica Semi-supervisada para la detección de genes ortólogos”, el centro de esta investigación lo constituyen los distintos enfoques de la técnica semi-supervisada.

En la actualidad se cuenta con información que han brindado estudios realizados sobre la detección de ortólogos y la anotación funcional de proteínas. Por otra parte el aprendizaje semi-supervisado tiene un enorme valor práctico y en muchas tareas, existe escasez de datos etiquetados, las etiquetas pueden ser difíciles de obtener porque requieren expertos humanos, dispositivos especiales o experimentos costosos y lentos. El objetivo del trabajo es aprovechar esta información disponible en el proceso de aprendizaje para detectar genes ortólogos de las especies Chlamydia trachomatis y Chlamydophila pneumoniae.

Se realizaron diferentes experimentos con el algoritmo Co-agrupamiento espectral unido al no supervisado KMeans y al semi-supervisado MPCKMeans tomando como base los resultados del método BBH. Los resultados no fueron los esperados lo que indica que quizás el enfoque probabilístico con restricciones no sea la vía más adecuada para la detección de ortólogos en secuencias genómicas.

Índice

Índice de Figuras

Introducción

La célula es la unidad morfológica y funcional de todo ser vivo, simple o complejo. De hecho es el elemento de menor tamaño que puede considerarse vivo**[1].**

La célula contiene en su interior el material genético que se compacta en un área discreta de la célula formando los cromosomas. A su vez en los cromosomas se disponen los genes ocupando una posición determinada**[2].** Un gen es una secuencia lineal organizada de nucleótidos en la molécula de ADN[1] (o ARN[2] en el caso de algunos virus), que contiene la información necesaria para la síntesis de una macromolécula con función celular específica, normalmente proteínas, pero también ARN.

En el estudio comparativo de los seres vivos, la homología es la relación que existe entre dos partes orgánicas diferentes cuando sus determinantes genéticos tienen el mismo origen evolutivo. En 1920 Alexander Weinstein acuñó el término genes homólogos para referirse a genes de especies distintas con expresiones fenotípicas[3] similares**[3].**

La secuencia de nucleótidos de un gen es transmitida de padres a hijos y es lo que principalmente cambia con la evolución. Cuando se examina el genoma de dos especies se espera encontrar genes equivalentes en ambas, con una secuencia algo diferente, más diferentes cuanto más remoto en el tiempo es el antepasado común. La expresión homología de secuencias se refiere a la correspondencia entre las cadenas de nucleótidos de esos dos genes, que es precisamente la que permite reconocer que son homólogos.

Dentro de la homología de secuencia se distinguen dos tipos de homología: la ortología y la paralogía. Los genes ortólogos conservan su homología (similitud de sus secuencias) y función a través de las diferentes especies partiendo de un ancestro

[1] Ácido desoxirribonucleico: Macromolécula que codifica los genes de las células, bacterias y algunos virus.

[2] Ácido ribonucleico: es normalmente el producto de la transcripción de un molde de ADN.

[3] Definición por el diccionario de Biotecnología-Glosario: Conjunto de todas los caracteres aparentes expresados por un organismo, sean o no hereditarias.

común. Existen además genes parálogos, que son aquellos que se encuentran en el mismo organismo, y cuya semejanza revela que uno procede de la duplicación del otro. La ortología requiere que se haya producido especiación, mientras que esta no es necesaria en el caso de la paralogía, que puede producirse sólo en los individuos de una misma especie.

La identificación de las relaciones de ortólogos entre diferentes organismos permite la anotación funcional de proteínas de un organismo para ser transferidos a su ortólogo. Este conocimiento permite validar las predicciones biológicas y evaluar el poder de los métodos desarrollados. En la actualidad estos datos proporcionan valiosa información para la comparación de genomas.

Por ejemplo, el estudio del cáncer ha prosperado por el estudio de modelos de ratón, y ha dado lugar a la aplicación médica en seres humanos. Cepas mutantes pueden ser aisladas que contienen defectos específicos en los genes que dan lugar a fenotipos de la enfermedad. Cruzamientos controlados se pueden utilizar para restaurar las funciones perdidas o inhibir genes en determinadas etapas del desarrollo y estudiar sus efectos sobre el organismo**[24]**.Lo que resulta de gran importancia para las ciencias médicas y las relacionadas con la vida en forma general. Existen hoy numerosas técnicas computacionales desarrolladas para este fin, con el objetivo de ampliar los conocimientos que se tienen hasta el momento y lograrlo en menor tiempo.

El problema de la detección de ortólogos usualmente se trata como un problema no supervisado donde se agrupan los genes ortólogos según un conjunto de rasgos como la similitud de las secuencias y su longitud, la conservación del orden alrededor de los genes, la similitud de la interacción entre proteínas en una determinada vecindad en sus respectivas redes de interacción y la distancia evolutiva.

El aprendizaje no supervisado es un método de aprendizaje automático donde un modelo es ajustado a las observaciones**[5]**. Se diferencia del aprendizaje supervisado por el hecho de que no hay un conocimiento a priori. Es adecuado para clasificar ejemplos cuyas clases se desconozcan o para comprender mejor la información de la que se dispone.

En un esquema de clasificación supervisada tradicional, si el entorno donde el clasificador ha sido entrenado sufre algunas variaciones, o si llegan a surgir nuevas clases no consideradas en el conjunto de entrenamiento , se requerirá que el clasificador sea nuevamente entrenado, por lo cual se hará necesario recurrir nuevamente al experto humano para que reconstruya el conjunto de entrenamiento, situación que en muchos casos resulta sumamente problemática por la dificultad y el costo que ello implica. Lo que sí resulta mucho más fácil, es obtener muestras no etiquetadas, razón por la cual se hace necesario diseñar métodos de aprendizaje que permitan utilizar tanto muestras etiquetadas como no etiquetadas**[6]**.

Este tipo de aprendizaje recientemente surgido recibe el nombre en la literatura científica de aprendizaje semi-supervisado o parcialmente supervisado. El aprendizaje semi-supervisado trata este problema usando una cantidad grande de datos sin etiqueta, junto con un conjunto (probablemente pequeño) de datos etiquetados, para construir clasificadores mejores. Su principal ventaja es que requiere menos esfuerzo humano y aprovecha además conocimiento previamente disponible, por lo cual se tiene gran interés en el ámbito científico tanto en aspectos teóricos como prácticos**[6].**

Existen algunos software de comparación de genomas y detección de ortólogos (Inparanoid/Multiparanoid, COG/KOG, OrthoMCL, HomoloGene) que estudian la homología mediante técnicas de alineación y otros basados en la ubicación de genes estudian los bloques de orden conservado compartidos entre genomas **[4]**,los software antes mencionados muestran un por ciento de precisión cerca del 90% y aunque es un por ciento bastante elevado continua siendo un problema potencial de investigación dentro del campo de la genómica comparativa.

Intentar beneficiarse de la información que se conoce, como por ejemplo genes confirmados como ortólogos entre organismos o la función de proteínas conocidas es todavía un campo reciente pero promete buenos resultados, pues la técnica semi-supervisada ha sido empleada en problemas de clasificación donde existía información a priori que antes no se aprovechaba y los resultados han sido mejorados considerablemente. En la actualidad existen varias bases de datos disponibles a través

de la Web que almacenan los resultados de algoritmos de detección de genes ortólogos, por ejemplo, NCBI[4], euKaryotic Orthologous Group database (KOG) [7], OrthoMCL_DB [8], MultiParanoid [9], Eukaryotic Gene Orthologues database (EGO) [10], YOGY [11] y Roundup [12].

Problema a Resolver:

¿Cómo detectar genes ortólogos de las especies Chlamydia trachomatis y Chlamydophila pneumoniae, con el objetivo de aprovechar la información disponible sobre los genes ortólogos previamente detectados?

Se identifica como **objeto de estudio** el proceso de detección de genes ortólogos por lo que el **campo de acción** apunta a la detección de genes ortólogos de las especies Chlamydia trachomatis y Chlamydophila pneumoniae, utilizando un enfoque semi-supervisado.

Idea a defender

La aplicación de la técnica semi-supervisada a la información referente en secuencias genómicas de las especies Chlamydia trachomatis y Chlamydophila pneumoniae, permitirá la correcta detección de sus genes ortólogos.

Objetivo general

Aplicar un algoritmo de la técnica semi-supervisada para detectar genes ortólogos en las especies Chlamydia trachomatis y Chlamydophila pneumoniae.

Objetivos específicos

- Analizar la técnica semi-supervisada.
- Detectar genes ortólogos en las especies Chlamydia trachomatis y Chlamydophila pneumoniae empleando un algoritmo no supervisado y otro semi-supervisado.

[4]Sigla del ingles National Center for Biotechnology Information

- Comparar los resultados obtenidos entre cada uno de los algoritmos seleccionados.

El trabajo se estructura de la siguiente forma: introducción, tres capítulos cada uno de ellos con su introducción y conclusiones parciales, conclusiones, recomendaciones, referencias bibliográficas, bibliografía y anexos.

Capítulo 1 nombrado “Marco teórico de la investigación”. Donde se enuncian los principales conceptos de la genómica comparativa asociados al objeto de estudio y se explican los principales algoritmos empleados para la detección de genes ortólogos.

Capítulo 2 nombrado “La técnica semi-supervisada”: Donde se definen los principales conceptos asociados a la técnica de aprendizaje, se abordan los aspectos teóricos que se necesitan dominar en la investigación y se presentan algunos algoritmos que aplican la misma.

Capítulo 3 nombrado “Diseño del experimento y análisis de los resultados”: Donde se emplean algoritmos de diferentes técnicas de aprendizaje aplicados a la detección de genes ortólogos de las especies Chlamydia trachomatis y Chlamydophila pneumoniae y se comparan los resultados.

Capítulo 1 . Marco Teórico de la Investigación

En el presente capítulo se enuncian los principales conceptos referentes a la genómica comparativa vinculados al objeto de estudio. A continuación se justifica la selección del organismo modelo. Finalmente se caracteriza la técnica actual empleada para la selección de genes ortólogos y se analizan los principales algoritmos empleados para la detección de los genes ortólogos en diferentes especies.

1.1 Conceptos asociados al dominio del problema

Gen: Segmento de ADN (ARN en algunos virus) que constituye la unidad hereditaria del ser vivo. Funcionalmente es el segmento de ácido nucleico con información precisa para dirigir la síntesis de una cadena polipeptídica **[13]**.

Genoma: Es el conjunto de cromosomas que constituyen la dotación genética de una célula haploide. El Proyecto Genoma Humano tiene como objetivo analizar la secuencia del ADN humano, distribuido entre los 23 pares de cromosomas **[13]** .

Cromosoma: Componente de las células, de estructura filamentosa, portador de los factores de la herencia o genes. Se hallan en número constante, que en la especie humana, es de 22 pares más dos cromosomas sexuales, en total 46 cromosomas. Los cromosomas son muy visibles en el núcleo celular durante la mitosis **[14].**

Código genético: Se trata de la correspondencia entre las cuatro bases que constituyen el ADN, las complementarias del ARNm y los 20 aminoácidos que forman las proteínas **[13].**

Locus: es una posición fija sobre un cromosoma, como la posición de un gen(localización) o de un biomarcador (marcador genético) **[15]**.

Bioinformática: según una de sus definiciones más sencillas, es la aplicación de tecnología de computadoras a la gestión y análisis de datos biológicos. El término bioinformática hace referencia a campos de estudios interdisciplinarios muy vinculados, que requieren el uso o el desarrollo de diferentes técnicas que incluyen informática, matemática aplicada, inteligencia artificial, química y bioquímica para solucionar

problemas, analizar datos, o simular sistemas o mecanismos, todos ellos de índole biológica, y usualmente (pero no de forma exclusiva) en el nivel molecular.

El núcleo principal de estas técnicas se encuentra en la utilización de recursos computacionales para solucionar o investigar problemas sobre escalas de tal magnitud que sobrepasan el discernimiento humano. Los principales esfuerzos de investigación en estos campos incluyen el alineamiento de secuencias, la predicción de genes, montaje del genoma, alineamiento estructural de proteínas, predicción de estructura de proteínas, predicción de la expresión génica, interacciones proteína-proteína, y modelado de la evolución **[16]**.

Gen diana: Blanco biológico, es el gen que produce una determinada proteína a la cual se le quiere inhibir su función. Una proteína, enzima, receptor u otra molécula que puede desempeñar un rol en un proceso de enfermedad particular. Es sobre el cual se focalizará el descubrimiento y las estrategias terapéuticas **[17]**.

BLAST:(Basic Local Alignment Search Tool) es la principal aplicación para comparar secuencias. Las principales versiones son BLASTP y TBLASTN. BLASTP compara la secuencia de una proteína con una base de datos de proteínas **[18]**.

Método Wrapper: Seleccionan los atributos en función de la calidad del modelo de minería asociado a los atributos utilizados.

La extracción de atributos puede ser vista como una proyección del espacio de estudio ya que permite transformar el espacio de atributos, obteniendo otro espacio de atributos que represente la misma información de diferente manera.

Matriz Sparse: se dice que una matriz es sparse cuando el número de elementos no nulos es del orden de *n* siendo *n* el número de elementos de la matriz**[19]**.

La finalidad del uso de las matrices sparse es triple. Primero ahorrar la enorme cantidad de memoria desperdiciada almacenando una matriz llena de ceros; segundo, definir métodos para realizar todas las operaciones disponibles en una matriz pero en el caso de almacenamiento sparse y por último que los errores de resolución no sean los equivalentes a resolver una matriz con n elementos sino una con n. Resumiendo,

aprovechar la forma de la matriz para obtener todas las propiedades beneficiosas que se pueda.

1.2 Genómica comparativa

La genómica comparativa estudia las semejanzas y diferencias entre genomas de diferentes organismos, tiene como objetivo predecir la función de los genes a partir de su secuencia o de sus interacciones con otros genes. Es un intento de beneficiarse de la información proporcionada por las firmas de la selección natural para entender la función y los procesos evolutivos que actúan sobre los genomas. Aunque es todavía un campo reciente, promete adquirir nuevas percepciones sobre muchos aspectos de la evolución de las modernas especies. La cantidad total de información contenida en los genomas más complejos (750 megabytes en el caso del ser humano) requiere de la automatización de los métodos de la genómica comparativa. La predicción de genes es una aplicación importante de la genómica comparativa, como también lo es el descubrimiento de nuevos y no codificantes, pero funcionales, elementos del genoma.

La genómica comparativa se aprovecha tanto de las similitudes como de las diferencias en las proteínas, ARN, y regiones reguladoras de diferentes organismos para inferir cómo la selección natural ha actuado sobre tales elementos. Aquellos elementos que son responsables de similitudes entre diferentes especies se conservarían a través del tiempo (selección estabilizadora), mientras que los elementos responsables de las diferencias entre especies deberían divergir (selección direccional). Finalmente, aquellos elementos que no son importantes para los sucesos evolutivos del organismo no serán conservados (la selección es neutral). La identificación de los mecanismos de la evolución del genoma eucariota mediante genómica comparativa es uno de los objetivos importantes en esta área. Sin embargo, a menudo es complicado dada la multiplicidad de eventos que han tenido lugar a través de la historia de linajes individuales, dejando sólo trazas distorsionadas y superpuestas en el genoma de cada organismo vivo. Por esta razón, los estudios por genómica comparativa de pequeños organismos modelo (la levadura, por ejemplo) son de gran importancia para avanzar en el conocimiento de los mecanismos generales de la evolución.

Los genes homólogos **posibilitan** el reconocimiento de funciones comunes en distintos organismos, ya sea gracias a los genes ortólogos (genes homólogos presentes en distintos organismos que codifican proteínas con la misma función y que han evolucionado mediante descendencia directa) o a los genes parálogos (genes homólogos presentes en un mismo organismo que codifican proteínas con funciones similares, pero no idénticas) **[20]**.

1.3 Homología vs. Similitud

"Similitud es la observación o medición de parecido y diferencia, independiente del origen de ese parecido. Homología significa, específicamente, que las secuencias y los organismos en los que están presentes, descienden de un ancestro común [...]"**[22]**

En sentido estricto, la homología se refiere únicamente a un origen común entre dos caracteres. Por tanto, dos secuencias son homólogas o no homólogas y no hay ninguna gradación intermedia.

Similitud, en cambio, es una medida del parecido entre dos secuencias que puede cuantificarse. Dos secuencias pueden ser muy similares y sin embargo no ser homólogas (así como las alas de un murciélago y de una mariposa parecen iguales, sin embargo no hay un ancestro común entre las mariposas y los murciélagos que tienen alas).**[22]** De la misma manera, dos secuencias homólogas pueden haber divergido mucho en la historia evolutiva, haciéndolas poco similares.

Debido a que se ha usado la palabra homología en el contexto de similitud en muchas publicaciones, la tendencia de los autores es usar los términos "ortólogas" y "parálogas" al referirse a secuencias con origen evolutivo común, que son más específicos. Dos secuencias son ortólogas si fueron adquiridas por descendencia vertical (por ejemplo, de madre a hijo) y son parálogas si están presentes en más de una copia en el mismo organismo y tuvieron el mismo origen**[22].**

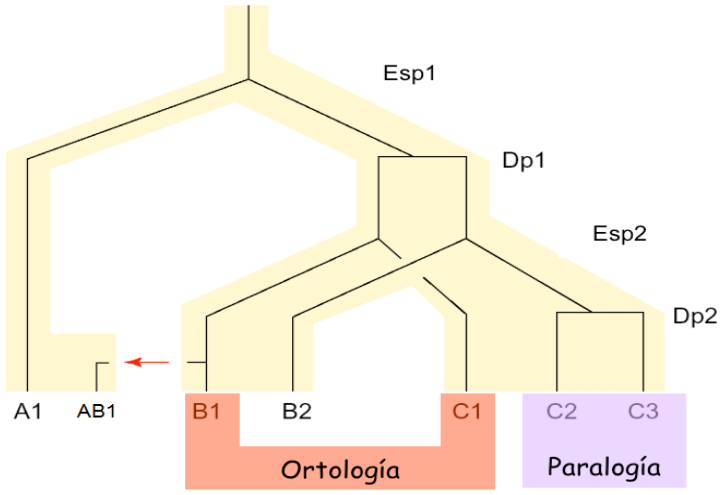

Ilustración 1. Evolución Idealizada de un Gen

En la figura anterior se muestran los dos tipos de Homología que existen: ortología y paralogía. Se muestra la evolución idealizada de un gen (líneas negras) a partir de un ancestro común, descendiendo hacia tres poblaciones A, B y C (fondo amarillo claro). Hay dos eventos de especiación (Esp1 y Esp2) en los puntos donde se forman las "Y" invertidas. También hay dos eventos de duplicación genética (Dp1 y Dp2) ilustrados como líneas horizontales. Dos genes cuyo ancestro común reside en la unión de una "Y" invertida son ortólogos (por ejemplo B1 y C1). Dos genes cuyo ancestro común reside en una línea horizontal son parálogos (por ejemplo C2 y C3). Los siete genes son homólogos entre sí porque proceden de un mismo ancestro común en la raíz del árbol**[23]**.

1.4 Biología molecular y el estudio de la vida

Entender las señales biológicas codificadas en el genoma constituye un desafío clave de la biología moderna. Estas señales se codifican en el alfabeto de cuatro nucleótidos del ADN y son responsables de todos los procesos moleculares en la célula. En particular, el genoma contiene el modelo de todos los genes codificadores de proteínas y el control de las señales utilizadas para coordinar la expresión de estos genes. El

correcto funcionamiento de cualquier célula se basa en el reconocimiento de estas señales y un gran número de mecanismos biológicos que han evolucionado hacia esta meta. Complejos de proteínas específicas son responsables de la copia de un segmento del gen de ADN al ARN mensajero (transcripción) y para su eventual traducción en proteína siguiendo el código genético para asignar un aminoácido a cada codón de tres nucleótidos.

Una clase específica de proteínas llamadas factores de transcripción son las que ayudan a la maquinaria de transcripción para con un gen diana y la unión de sus señales específicas de ADN (motivos de reglamentación) dar respuesta a las condiciones ambientales**[24]**. La información dentro de la célula guía estos procesos, utilizando interacciones proteína-proteína y proteína-ADN y otros mecanismos que aún no están bien estudiados.

La identificación de los genes de manera computacional sin embargo, sólo puede contar con la secuencia primaria del ADN del organismo. Los programas actuales utilizan las propiedades de la región codificante potencial de ADN. En particular, ya que los genes siempre comienzan con un (codón de inicio) ATG y terminan con TAG, TGA, o TAA (uno de los tres codones de parada), existen programas específicamente para buscar el codón de inicio y el de parada llamados ORFs (Marcos Abiertos de Lectura). El enfoque básico es identificar ORF que son demasiado largos y probablemente ocurren por casualidad**[24].**

Dado que los codones de parada se producen con una frecuencia de 3 en 64 de forma aleatoria, ORF de 60 o incluso 150 aminoácidos se producen con frecuencia por casualidad, pero ya ORF de 300 o miles de aminoácidos son casi siempre el resultado de la presión selectiva biológica**[24]**.

Por lo tanto, simples programas de cómputo pueden reconocer fácilmente los genes largos, pero muchos genes pequeños serán indistinguibles de ORFs falsas que surgen por casualidad. Esto se evidencia por el debate considerable sobre el número de genes, que cuenta con propuestas que van desde 4800 hasta 6400 genes. Esta situación es

peor en organismos superiores, genomas más complejos, como por ejemplo los mamíferos donde el número de propuesta va desde 30 hasta 120 miles de genes.

1.5 Organismo modelo

Los organismos más simples proporcionan excelentes modelos para desarrollar y probar los procedimientos necesarios para el estudio del genoma humano mucho más complejo. Hasta ahora, los genomas de más de 200 organismos han sido completamente secuenciados**[24]**. De particular interés para la investigación médica son las secuencias completas del genoma de organismos modelos, tales como bacterias, levaduras, hongos, gusanos, moscas y ratones, cada uno de estos enseña los diferentes aspectos de la biología humana. Proyectos de secuenciación del genoma de otros organismos modelo, como el chimpancé, también están cerca de su finalización.

1.6 Alineamiento de secuencias

Para identificar una relación evolutiva entre una secuencia recién determinada y una familia génica conocida debe evaluarse la cantidad de similitud compartida. La forma más simple de comparar dos secuencias es alinearlas insertando caracteres de hueco para hacer que estén en concordancia vertical. Contar las posiciones con caracteres coincidentes da una puntación simple para el alineamiento.

Los alineamientos son modelos que reflejan diferentes perspectivas biológicas. Un modelo no es por tanto más o menos correcto que otro**[25]**. Dos enfoques generales consideran la similitud (a) a través de toda la longitud de las secuencias y (b) a través de solo parte de las secuencias.

Los programas Fasta y BLAST son métodos de búsqueda de similitud local que se concentran en hallar alineamientos cortos idénticos, que pueden contribuir a un alineamiento global. Estos son los más usados en la mayoría de los algoritmos que existen para la detección de genes ortólogos.

El alineamiento completo (global) de dos secuencias (Smith-Waterman, swalign) es muy preciso**[25]** y garantiza obtener el alineamiento óptimo. Sin embargo este algoritmo es muy lento. El tiempo de cálculo es proporcional al producto de las longitudes de las dos secuencias que se quieren alinear (o al producto de una secuencia problema y todas las secuencias de la base de datos), existen otros como por ejemplo el (Needleman Wunsch, nwalign).

Los alineamientos basados en la secuencia o la estructura son ambos, modelos imperfectos, pues ninguno puede recoger todos los niveles de información biológica. Ambos enfoques son representaciones básicas de aspectos particulares de la biología.

BLAST es la principal aplicación para comparar secuencias. Las principales versiones son BLASTP y TBLASTN. BLASTP compara la secuencia de una proteína con una base de datos de proteínas. TBLASTN compara la secuencia de una proteína con una base de datos de nucleótidos.

1.7 Introducción a la Inteligencia Artificial.

La Inteligencia Artificial es una rama de la Ciencia de la Computación dedicada a la creación de hardware y software que imita el pensamiento humano. Su principal objetivo es llevar a la computadora las amplias capacidades del pensamiento humano y, para ello, se convierten a las computadoras en “entes inteligentes” con la creación de software que les permite imitar algunas de las funciones del cerebro humano en aplicaciones particulares. El fin no es reemplazar al hombre, sino proveerlo de una herramienta poderosa para asistirlo en su trabajo.

La IA se ocupa de la representación, adquisición y procesamiento de conocimientos en forma automatizada, de la arquitectura de los programas para estas actividades y de los lenguajes en los que se expresan los programas. La modelación computacional de los procesos cognoscitivos es también un área de interés de la IA, además se incluyen la percepción, la comprensión y síntesis del lenguaje natural, la robótica inteligente, la modelación del razonamiento, la programación automática y otras más, todas ellas de naturaleza no numérica y todavía del dominio de la heurística **[21]**.

Los programas de IA requieren de conocimiento y sus técnicas consisten en métodos para explotar este, el cual debe ser representado de manera que: Capte generalizaciones: No es una base de datos. No debe ser necesario representar cada situación individual, sino que se agrupen las situaciones que compartan propiedades importantes. Si no tiene esta característica se necesitaría más espacio del disponible y más tiempo del que tenemos para mantenerlo actualizado. Pueda ser comprendido por los especialistas que lo proporcionan. Deba ser modificable fácilmente. Pueda ser usado en muchas situaciones diversas, incluso si no es totalmente preciso o completo. Pueda ser usado para extenderse a sí mismo.

La IA incluye la solución de problemas dentro de diversos campos como la robótica, la comprensión y traducción de lenguajes, el reconocimiento y aprendizaje de palabras de máquinas o los variados sistemas computacionales expertos, que son los encargados de reproducir el comportamiento humano en una sección del conocimiento. Sus técnicas son muy aplicables a problemas de predicción, clasificación y reconocimiento de patrones.

1.8 Técnicas computacionales aplicadas a los genes ortólogos

En la actualidad las tareas de identificación de genes ortólogos se realizan de forma automática, mediante el empleo de herramientas computacionales diseñadas para este fin. Debido fundamentalmente a la demora que ocasionaría la clasificación de cada uno de los dominios de ortólogos de manera manual, teniendo en cuenta las grandes dimensiones que pueden alcanzar los genomas de las diferentes especies. Para identificar pares de proteínas con mayor similitud funcional posible, es importante la clasificación de ortólogos, donde se utiliza la información acerca de una proteína en una especie para la anotación funcional de la proteína en otras especies.

Estas herramientas desarrolladas pertenecen al área de la inteligencia artificial y emplean el aprendizaje automático para realizar sus tareas de clasificación.
El aprendizaje automático es el área de la inteligencia artificial que se ocupa de desarrollar algoritmos (y programas) capaces de aprender, y constituye, junto con la estadística, el corazón del análisis inteligente de los datos**[26]**.

En la tarea de clasificación, existen métodos que se aplican cuando los ejemplos tienen información de la clase a la que pertenecen, y métodos para la tarea de agrupamiento, en la que los ejemplos no contienen información de la clase a la que pertenecen, estos métodos también suelen llamarlos aprendizaje supervisado y no supervisado.

El Aprendizaje supervisado es un método que aprende a partir de un conjunto de instancias pre-etiquetadas para predecir la clase a que pertenece una nueva instancia y el Aprendizaje no supervisado es también un método de aprendizaje automático donde un modelo es ajustado a las observaciones. Está formado por una variable dependiente (clase), cuyo objetivo es averiguar dicha clase para casos nuevos**[5]**. Se distingue del aprendizaje supervisado por el hecho de que no hay un conocimiento a priori. Puede valer para clasificar casos cuyas clases se desconozcan o, simplemente, para comprender mejor la información de la que disponemos. Lo que si resulta mucho más fácil es la obtención de datos no etiquetados razón por la cual se hace necesario diseñar métodos de aprendizaje que permitan utilizar tanto muestras etiquetadas como no etiquetadas.

Tanto en la comparación como en la anotación de genomas desconocidos se estudian grupos de genes cercanos que conservan la organización local (orden y distancia) a través de la evolución. Estos grupos se denominan “bloques de orden conservado” (synteny blocks)**[4]**.

Algunos software de los que se emplean para este propósito estudian la homología mediante técnicas de alineación y otros basados en la ubicación de genes estudian los bloques de orden conservado compartidos entre genomas comparados sin tener en cuenta los reordenamientos globales de genomas **[23]**.

A continuación se presentarán algunas de las herramientas que existen con el objetivo de lograr la detección de genes ortólogos.

1.8.1 Inparanoid

Inparanoid fue desarrollado para encontrar todos los ortólogos de especies Y para un gen en las especies X. Tiene buenos resultados en el equilibrio de la tasa de falsos

negativos y falsos positivos, y es apropiado como una herramienta de propósito general en la ortología. El programa Inparanoid se desarrolló para identificar los grupos de ortólogos evitando al mismo tiempo la inclusión de parálogos**[27]** .

El algoritmo se ha ido mejorando a lo largo del tiempo y ahora es más específico. Se ha fichado como el mejor método de identificación de ortólogos en cuanto a la identificación de proteínas funcionalmente equivalentes. El enfoque en que se basa es de dos pasos de BLAST para hacer uso de la alta precisión en el ajuste de la matriz de composición, lo que evita el truncamiento de la alineación**[28]**.

Este algoritmo es más apropiado en el estudio de pares de proteínas con distancias evolutivas relativamente pequeñas. Los parámetros por defecto que propone son umbral = 0,5 y confianza = 0,05. Inparanoid opcionalmente permite utilizar un tercio del genoma como un grupo externo y así detectar las secuencias que faltan y mejorar la detección de ortólogos**[29].**

1.8.2 Bidirectional Best Hit

BBH se alza como uno de los mejores métodos, salvo para los casos teóricos donde dos proteínas de las especies A tienen el mismo puntaje a una proteína de la especie B o cuando se tiene en cuenta la fusión de los genes. Es el método más usado para determinar pares de ortólogos**[30]**.

El método requiere una fuerte relación entre los ortólogos.

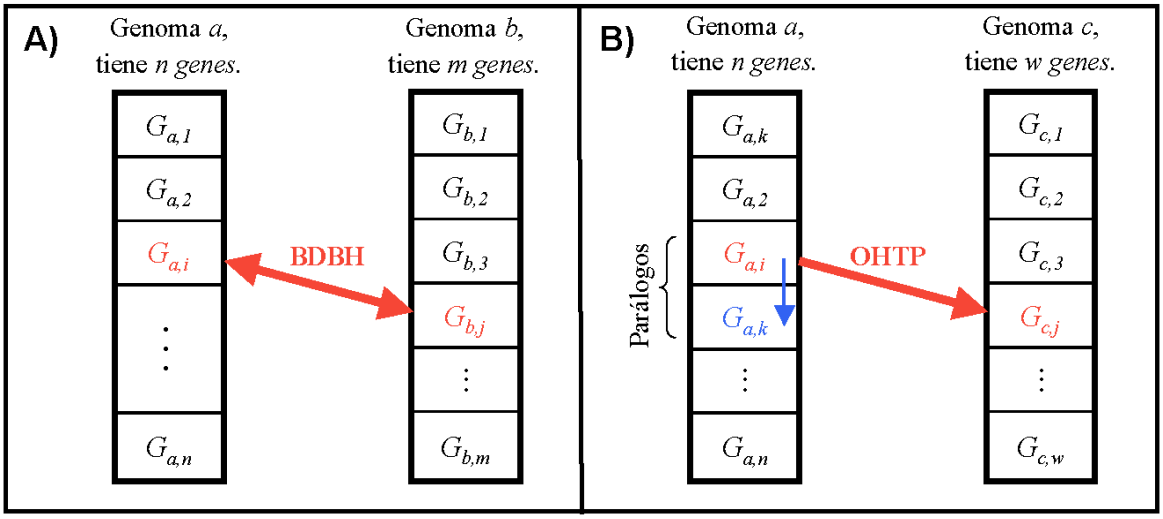

Ilustración 2. Identificación de probables ortólogos (PO)

A) Genes más parecidos bidireccionalmente (BDBH por sus siglas en Inglés Bi–Directional Best Hits)**[31]**. Si el genoma a tiene n genes y el b tiene m, el gene Ga,i será BDBH de Gb,j (genes en rojo) si al comparar Ga,i contra todos los m genes en b, Gb,j es el más parecido; y viceversa. Esta relación se representa con la flecha roja de dos puntas.

B) Mejor ortólogo que parálogo (OHTP por sus siglas en Inglés Ortholog Higher Than Paralog)**[31]**. Cuando no se encontró un BDBH para Ga,i en un genoma **c**, se toma como PO al gen más parecido en **c**, Gc,j (flecha roja unidireccional vinculando a los dos genes rojos), siempre y cuando no existiera en el genoma **a** un gene Ga,k (gen azul) con mayor parecido a Ga,i que Gc,j (flecha azul relacionando a dos genes dentro del genoma a). En este caso Ga,i es OHTP de Gc,j.

El grueso de las flechas roja y azul es proporcional al parecido entre los genes respectivos.

1.8.3 COG (Clusters of Orthologous Groups)

Originalmente creado por Tatusov en 1997 y actualizado por David Kristensen. En esencia este algoritmo permite definir grupos de secuencias (potencialmente) ortólogas entre 3 o más genomas a partir de los resultados de BLAST de todos contra todos. Para

ello utiliza el concepto de los 'bidirectional best hits'(BBH), que se dibuja como flechas bidireccionales. El procedimiento COG presta especial atención a las proteínas con múltiples dominios **[30]**.

1.8.4 OrthoMCL

El algoritmo MCL comienza con un estudio de similitud general entre las secuencias, BLASTP. Después de lo cual los mejores pares de similitud entre las especies se marcan como posibles ortólogos y los pares de similitud recíproca como parálogos recientes. Una matriz de similitud se calcula, seguida de una agrupación de Markov[5], que determina los grupos de ortólogos**[32]**.

MCL encuentra las familias de genes en un genoma y luego los genomas restantes se alinean con este genoma "ancla" localizando bloques en los genomas estudiados. El algoritmo posee dificultades, existen casos donde dos o más familias pueden quedar unidas equivocadamente o una familia puede ser separada de forma errada.

1.8.5 HomoloGene

HomoloGene es un sistema para la detección automatizada de los genes homólogos anotados entre varios genomas eucariotas completamente secuenciados. Los conjuntos de genes de partida se derivan de la anotación de genes de más de una docena de los genomas eucariotas secuenciados. El proceso global del agrupamiento de genes es guiado por un árbol de la evolución. Después de esto, se utilizan medidas de similitud para identificar las relaciones antiguas y las familias ampliadas del gen**[33]**.

HomoloGene proporciona la estructura de la proteína conservada. Además, se incluye información complementaria, cuando sea posible, proporcionando enlaces a la información fenotípica y a la literatura publicada.

Este algoritmo puede ser utilizado en programas para la detección de ortólogos donde parte de un archivo que contiene apareamientos de genes ortólogos, y selecciona

[5] Modelo estadístico en el que se asume que el sistema a modelar es un proceso de parámetros desconocidos.

aquellas líneas que contienen genes de las 2 especies, los cuales se pretenden alinear**[34]**.

1.8.6 UPGMA (Unweighted Pair Group Method using Arithmetic averages).

Método para análisis filogenético definido por Peter H. A. Sneath y Robert R. Sokal en 1973. Es un algoritmo **[35]** heurístico que usualmente encuentra una solución muy acertada. Consiste en la búsqueda de la distancia más pequeña en la matriz de distancias genéticas y agrupa las unidades que la conforman como una sola unidad taxonómica independiente. Se calculan los promedios de la nueva unidad contra las restantes creando una nueva matriz y se repite el proceso hasta que todas las unidades queden unidas a un único elemento (ancestro hipotético). El método UPGMA al igual que todos los demás métodos filogenético basados en distancia, dadas las secuencias genéticas de los taxones[6], genera una matriz de distancia y realiza sus cálculos con esta. La matriz de distancia es una matriz cuadrada de tamaño n, donde n es el número de taxones a clasificar y es la distancia genética entre una especie i y una especie j.

1.8.7 DomClust

Este método toma como entrada todos los datos con similitud y clasifica los genes sobre la base del agrupamiento del algoritmo jerárquico tradicional UPGMA. El procedimiento posterior divide los árboles resultantes de tal manera que dentro de una especie de genes parálogos se dividen en grupos diferentes a fin de crear grupos de ortólogos plausibles. Como consecuencia, el procedimiento puede dividir genes en los mínimos dominios necesarios para la agrupación de ortólogos. El procedimiento, llamado DomClust, se puso a prueba con la base de datos de COG como una referencia. Al comparar varios algoritmos de agrupamiento junto con los mejores criterios convencionales del (BBH), se mostró que este método dio una mejor clasificación de acuerdo con el COG. Mediante la comparación de los resultados generados por la agrupación de conjuntos de datos de diferentes versiones, también se

[6] Unidad sistemática que designa un nivel jerárquico en la clasificación de los seres vivos, como la especie, el género, la familia, el orden y la clase.

encontró que este método mostró una estabilidad relativamente buena en comparación con los métodos basados en BBH[36].

Este algoritmo fue desarrollado originalmente para la base de datos comparativa del genoma microbiano (MBGD) [7]**[36]**, es una extensión natural del algoritmo tradicional de agrupamiento jerárquico y es adecuado para dividir los genes en los ámbitos mínimamente necesarios para la agrupación de ortólogos.

Fue implementado en el lenguaje de programación C. El programa fue validado utilizando la base de datos COG como referencia.

1.8.8 BranchClust

La mayoría de los métodos conocidos utilizados para la detección de ortólogos se basan en la similitud y la secuencia específica del genoma. Los métodos filogenéticos son más fiables pero son difíciles de implementar y la complejidad aumenta con el aumento del número de taxones de que se trate**[31]**.

Este algoritmo que se describe aquí para la selección automatizada de ortólogos reconoce genes ortólogos de diferentes especies en un árbol filogenético para cualquier número de taxones. El BranchClust se basa en la búsqueda de similitud de secuencia utilizando BLAST como instrumento primordial. El algoritmo que se propone aquí automatiza completamente el proceso de montaje de las familias de genes para un determinado número de taxones**[31]**. Analiza los árboles para delimitar las familias de ortólogos dentro de una superfamilia que contiene varias familias de genes parálogos. La idea subyacente es que los genes estrechamente relacionados se encuentran en una rama que salen de un nodo a un árbol, por lo que la tarea de detectar las familias de diferentes taxones *n* es simplemente una tarea de detectar las ramas que contienen grupos de genes de todos, o casi todas, las especies .

Para seleccionar todos los posibles ortólogos realiza un BLAST *n* de todos los genomas contra la base de datos compuestas por *n* genomas, y luego ensambla todos sus éxitos para cada gen de cada genoma en superfamilias.

[7] Sigla del inglés microbial genome database

BranchClust proporciona un sistema automatizado, con un enfoque sólido para reunir las familias de ortólogos. El uso de BranchClust permitirá incluir a una mayor parte de los genomas completamente secuenciados basado en reconstrucciones filogenéticos **[31]**.

1.8.9 SOAR

Algunos software como el SOAR **[4]** utilizan heurísticas de reordenamiento global adicionales al tratamiento de la homología para estimar la distancia evolutiva entre las especies comparadas. SOAR comienza identificando las familias de genes homólogos y la correspondencia entre estas familias en los dos genomas comparados utilizando búsqueda de homología. Las familias son tratadas como copias de los genes y la asignación de ortólogos se formula como el problema de optimización de reordenar un genoma como una secuencia de genes posiblemente duplicados en otro con el mínimo número de inversiones.

El proceso de reordenamiento debe conducir a los pares de ortólogos. Dicho problema consiste específicamente en calcular la distancia de inversión con signo y con duplicados (SRDD) entre dos genomas. Utiliza las técnicas de partición común mínima y descomposición máxima de ciclos en un grafo completo.

SOAR **[4]** no tiene en cuenta los inparalogs [8]es decir, asume que no hubo duplicaciones de genes después de la especiación. Devuelve las relaciones uno a uno de ortólogos y todos los genes son forzados a formar pares de ortólogos. Trabaja con genomas de un solo cromosoma.

1.8.10 MSOAR

El MSOAR a diferencia del SOAR **[4]** se basa en un enfoque de máxima parsimonia que combina los eventos de reordenamiento incluyendo inversiones, translocaciones, fusiones y fisiones de genes con los eventos de duplicación. En este caso se minimiza el número de eventos calculando la distancia de reordenamiento/duplicación. MSOAR sí procesa genomas con múltiples cromosomas y realiza mejoras a las técnicas

[8] Genes ortólogos que sufrieron duplicación en una especie

empleadas en SOAR. Incorpora un nuevo paso de post-procesamiento para eliminar pares de genes “con ruido” que son más propensos a constituir inparalogs.

En una experimentación recogida en (Fu et al., 2007) **[4]** con el genoma del ratón y el hombre se comprobó que la gran mayoría de los resultados de MSOAR están acorde con la información de los bloques de orden conservado entre estas especies. La experimentación con datos simulados que aparece en (Fu et al., 2007) demuestra que con una alta probabilidad (>90% para MSOAR) identifica los ortólogos principales e inparalogs.

1.8.11 RSD (Reciprocal Smallest Distance)

Este algoritmo llamado el recíproco de distancia más pequeña (RSD) **[31]**, mejora el procedimiento común de tomar mejores éxitos recíprocos (RBH[9]) en la identificación de ortólogos mediante la alineación global de secuencias y la estimación de máxima verosimilitud de distancias evolutivas para detectar ortólogos entre dos genomas. RSD encuentra muchos posibles ortólogos desconocidos por RBH porque es menos probable que cometa errores en la presencia de parálogos cerca en los genomas. El RSD ofrece una gran flexibilidad en la investigación de valores de los parámetros que pues permite al usuario buscar cada vez ortólogos más distantes entre especies muy divergentes, entre otras ventajas. La flexibilidad de esta herramienta hace que sea una adición única y de gran alcance a otros enfoques para la detección de ortólogos.

El algoritmo de RSD original ya ha sido utilizado en una variedad de contextos para aumentar la comprensión de la naturaleza de la organización del genoma e identificar las tendencias filogenéticas entre especies diferentes.

En un esfuerzo de corregir algunos errores que existen en otros algoritmos de detección de ortólogos, se desarrolla este nuevo algoritmo. Este enfoque, llamado el recíproco de distancia más pequeña (RSD)**[31]**, ha demostrado que proporciona listas más completas de ortólogos que otros métodos.

[9] Sigla del ingles reciprocal best hits

1.8.12 Algoritmo BUS (Best Unambiguous Subset)

Un BUS es un conjunto de genes que pueden ser aislados como un grupo de homología mientras preserve todas las conexiones que potencialmente representen relaciones de ortólogos. Este algoritmo **[4]** considera las mejores correspondencias como conjunto de genes. Permite la aplicación recursiva del agrupamiento de genes en subconjuntos más pequeños reteniendo las ambigüedades hasta que aparezca más información como el orden conservado de los genes entre ortólogos consecutivos que facilita el reconocimiento de genes vecinos.

En el algoritmo la similitud entre los genes se representa como un grafo bipartido [10]que conecta genes de dos especies. Este grafo se construye a partir de "hits" obtenidos con **BLAST [4]** considerando correspondencias hacia delante y hacia atrás. Cada arista se pesa con la similitud de la secuencia de aminoácidos y la longitud de la correspondencia. Como pre-procesamiento eliminan las aristas que pesan menos del 80% de la arista de mayor peso en similitud y longitud. Luego construyen los bloques de orden conservado para mantener las correspondencias preferentemente dentro de estos bloques. Más adelante buscan conjuntos de genes localmente óptimos de modo que todas las mejores correspondencias dentro de estos conjuntos se mantengan dentro de los mismos y no existan genes fuera del conjunto que tenga correspondencias dentro del conjunto. El grafo bipartido que se construye es máximamente separable mientras mantiene todas las posibles relaciones entre ortólogos. Finalmente, el algoritmo devuelve las componentes conexas de este grafo que contienen pares de ortólogos y grupos de homología.

1.8.13 Algoritmo basado en BUS

Es un algoritmo que realiza la conformación de los grupos de homología entre los genomas con información sobre la homología de las secuencias, bloques de orden

[10] Se denomina en Teoría de grafos a un grafo no dirigido cuyos vértices se pueden separar en dos conjuntos disjuntos V1 y V2 y las aristas siempre unen vértices de un conjunto con vértices de otro.

conservado y reordenamientos globales. Empleando el algoritmo Maule **[4]** para obtener los reordenamientos globales en los genomas.

Este algoritmo formula el problema de la correspondencia entre genes en el contexto de la teoría de grafos. Representando las similitudes entre genes como un grafo bipartido y conectando los genes entre las dos especies. Cada arco es pesado de acuerdo con una medida de distancia según los tres rasgos que caracterizan a estos genes: su secuencia de aminoácidos, longitud de secuencia y pertenencia a regiones verdaderamente homólogas. La medida de distancia es calculada mediante una versión modificada de la distancia euclidiana **[4]**.

1.8.13.1 Algoritmo de Mauve

Este algoritmo analiza múltiples réplicas al mismo tiempo **[4]**, funciona con el método de encontrar semillas y extenderlas. Al igual que otros métodos de alineación, este algoritmo usa una técnica de alineación anclada para una rápida alineación de los genomas. El Mauve es un algoritmo que, durante el proceso de alineación, identifica los segmentos conservados que aparentan no haber sido alterados por el reordenamiento. Estas regiones son referidas como Locally Collinear Blocks (LCB).

Mauve es más práctico para cuando se está comparando genomas que son estrechamente cercanos en la cadena evolutiva. A diferencia de muchos, permite que el orden de las anclas alineadas sea reconfigurado en cada genoma permitiendo así la identificación de genomas reordenados. Una vez que el conjunto de anclas alineadas que definen los LCB ha sido determinado, Mauve completa la alineación global con fisuras o huecos para cada LCB, aplicando el algoritmo progresivo de alineación global ClustalW. Aunque cuando existen muchos segmentos duplicados es muy difícil la alineación de genomas. Este algoritmo original no funciona bien para grandes números de categorías porque no puede alinear regiones conservadas a través de los subconjuntos de los genomas comparados, muestra regiones que son compartidas por todas las secuencias y estás comparaciones pueden tomarse horas de cálculo.

1.8.13.2 Algoritmo progresivo de Mauve

El Mauve progresivo emplea una metodología algorítmica diferente para evaluar alineaciones de segmentos conservados a través de los subconjuntos de categorías. Este nuevo algoritmo puede ser aplicado a un número mayor de genomas que al algoritmo original, posibilita alinear genomas más divergentes y genomas con menos del 50% de nucleótidos afiliados. El ajuste manual de los parámetros de evaluación para la alineación no es usualmente necesario. Aunque es sustancialmente más lento y consume mucho más memoria que el Mauve original **[4]**.

1.9 Conclusiones

El estudio de la genómica comparativa a pesar de ser un campo muy reciente ha contribuido al avance de distintos campos de la ciencia como son la medicina, la agricultura, y la biotecnología; gracias al descubrimiento de secuencias de genes necesarias para la producción de proteínas de importancia médica y a la comparación de secuencias genómicas de distintos organismos. La genómica es hoy de gran importancia en la investigación de enfermedades genéticas, además ayudará al entendimiento de la evolución y a la funciones del genoma humano.

El primer objetivo en la genómica comparativa es la determinación de la correspondencia correcta de segmentos de cromosomas y elementos funcionales a través de las especies comparadas, lo que involucra el reconocimiento de segmentos ortólogos de ADN. Existen hoy herramientas computacionales para lograr este acometido las cuales aunque presentan buenos resultados no logran tener en cuenta en el proceso de clasificación toda la información de la que se dispone.

Capítulo 2 . La Técnica Semi-supervisada

En este capítulo se definen los principales conceptos asociados a la técnica de aprendizaje, se abordan los aspectos teóricos que se necesitan dominar en la investigación y se presentan algunos algoritmos que aplican la misma.

2.1 Aprendizaje Semi-supervisado

El aprendizaje Semi-Supervisado (SSL) tiene mitad del aprendizaje supervisado y mitad del no supervisado **[37]**. Además que adiciona datos no etiquetados, los algoritmos siempre tienen alguna información supervisada pero no necesariamente para todos los ejemplos. SSL es visto como aprendizaje no supervisado guiado por las restricciones. Por el contrario, la mayoría de otros enfoques ven SSL como aprendizaje supervisado con información adicional sobre la distribución de los x ejemplos. La última interpretación parece estar más en consonancia con la mayoría de las aplicaciones, donde el objetivo es el mismo que en el aprendizaje supervisado. Sin embargo, esta visión no se aplica fácilmente si el número y la naturaleza de las clases no son conocidos de antemano, sino que tienen que deducirse de los datos.

El aprendizaje semi-supervisado será más útil cuando existan muchos más datos no etiquetados que etiquetados. Esto es probable que ocurra si la obtención de los puntos de datos es fácil, porque adquirir las etiquetas lleva mucho tiempo, esfuerzo, o dinero. Este es el caso en muchas áreas de aplicación del aprendizaje de máquina por ejemplo en reconocimiento de voz, que cuesta poco grabar grandes cantidades de expresión, pero etiquetarlo requiere que algún humano lo escuche y lo transcriba **[38]**.

2.2 ¿Los datos no etiquetados mejoran o degradan el rendimiento de la clasificación?

Es razonable esperar una mejora en el rendimiento de la clasificación para cualquier aumento en el número de muestras (etiquetado o no etiquetado): más datos, mejor. De hecho, la literatura existente presenta resultados empíricos que atribuyen el valor positivo a los datos no etiquetados. La declaración de O'Neill **[38]** que "no deben desecharse las observaciones no clasificadas ciertamente" (O'Neill, 1978) parece ser

confirmada por los estudios teóricos, más notablemente por Castelli (1994), Castelli y Tapa (1995, 1996), y Ratsaby y Venkatesh (1995).

La esencia de estas investigaciones teóricas es que teniendo en cuenta los parámetros y el modelo se asegura que se tendrá una reducción esperada en el error de la clasificación mientras más datos sean reunidos (etiquetado o no etiquetado). Es más, los datos etiquetados son exponencialmente más eficaces reduciendo el error de la clasificación que los datos no etiquetados.

Sin embargo, un análisis más detallado de resultados empíricos actuales revela algunos aspectos enigmáticos de los datos no etiquetados. Por ejemplo, Shahshahani y Landgrebe (1994) **[38]** publicaron un informe de los experimentos dónde los datos no etiquetados degradaron el rendimiento de los clasificadores de Bayes con las variables Gaussianas.

Ellos atribuyen estos casos a las desviaciones de los supuestos del modelo, tales como los valores extremos y "muestras de clases desconocidas" Que incluso sugieren que las muestras sin etiqueta se deben utilizar con cuidado y sólo cuando los datos etiquetados pueden producir un clasificador pobre.

Otro ejemplo representante de ello es la investigación de Nigam et al. (2000) **[38]** sobre la clasificación de texto, donde los clasificadores a veces muestran una degradación del rendimiento. Se sugieren varias posibles fuentes de dificultades: problemas numéricos en el algoritmo de aprendizaje, los desajustes entre los grupos naturales en el espacio de características y las etiquetas actuales.

2.3 Aprendizaje semi- supervisado inductivo vs transductivo

En la actualidad existen dos entornos diferentes de aprendizaje semi-supervisado**[39]**: el aprendizaje semi-supervisado inductivo y transductivo. En la clasificación supervisada con todos los ejemplos de entrenamiento etiquetados, siempre se está interesado en la exactitud de la clasificación de futuros datos de prueba. En la clasificación semi-supervisada, sin embargo, el conjunto de entrenamiento contiene algunos datos no etiquetados, y dos objetivos diferentes son posibles.

• (inductivo) Predecir las etiquetas de los datos de prueba: Dado un ejemplo de entrenamiento, el aprendizaje semi-supervisado inductivo aprende de una función f: $X \rightarrow Y$ por lo que f se espera que sea un buen clasificador para los datos futuros. Al igual que en el aprendizaje supervisado, se puede estimar la exactitud de la clasificación de datos futuros mediante el uso de una muestra de prueba por separado que no está disponible durante el entrenamiento.

• (transductivo) Predecir las etiquetas de los casos sin etiquetar en la muestra de entrenamiento: Dada una muestra de entrenamiento, el aprendizaje transductivo entrena una función f que se espera que f sea un buen clasificador de los datos sin etiqueta. Donde f solo es definido en el ejemplo de entrenamiento dado, y no es requerido para hacer predicciones fueras. Por tanto, es una función simple.

Existe una analogía interesante: aprendizaje semi-supervisado inductivo **[39]** es como un examen en clase, donde las preguntas no son conocidos de antemano, y un estudiante tiene que prepararse para todas las preguntas posibles, en cambio, el aprendizaje transductivo es como para llevar a casa el examen, donde el alumno sabe las preguntas del examen y no necesita preparar más allá de aquellos.

2.3.1 Self-Training

Self-Training **[40]** es una técnica de uso común para el aprendizaje semi-supervisado. Este algoritmo es caracterizado por ser el más simple y fácil para aplicar las técnicas de aprendizaje semi-supervisado por el hecho de que el proceso de aprendizaje utiliza sus propias predicciones para enseñarse a sí mismo. En Self-Training un clasificador es entrenado primeramente con un una pequeña cantidad de datos etiquetados. El clasificador se usa para clasificar los datos no etiquetados. Por lo general puntos no etiquetados más seguros, se juntan con sus etiquetas previstas, y son añadidas al conjunto de entrenamiento. El clasificador es re-entrenado y se repite el procedimiento durante un cierto número de iteraciones o hasta que algunos criterios de terminación se cumplan. El clasificador utiliza sus propias predicciones para aprender por sí mismo. El procedimiento también es llamado self-teaching o bootstrapping (que no debe confundirse con el procedimiento estadístico con el mismo nombre).

2.4 Modelos Generativos

Una formulación natural del aprendizaje semi-supervisado es la utilización de modelos probabilísticos generativos **[39]**. Los datos sin etiqueta dicen cómo las instancias de todas las clases, se mezclan entre sí, y se distribuyen. Si es conocido cómo las instancias de cada clase se distribuyen, se puede descomponer la mezcla en clases individuales. Esta es la idea detrás de los modelos de mezcla para el aprendizaje semi-supervisado.

Suponiéndose que se conoce que (o se asume) los datos proceden de dos distribuciones Gaussianas, pero que no se conocen sus parámetros (la media, varianza, y las probabilidades a priori), los datos pueden usarse (con la etiqueta y sin etiqueta) para estimar estos parámetros para ambas distribuciones. Los datos etiquetados en realidad pueden ser engañosos. Los datos no etiquetados sin embargo, ayudan a identificar las medias de las dos distribuciones Gaussianas. Computacionalmente, se seleccionan los parámetros para maximizar la probabilidad de generar datos de entrenamiento a partir del modelo propuesto. Si sólo se han etiquetado los datos, la búsqueda de este conjunto de parámetros es sencillo, y la estimación de máxima verosimilitud (EMV) a menudo se puede calcular en forma cerrada.

Los modelos generativos permiten modelar explícita o implícitamente la distribución de las entradas como las salidas; y a partir de un muestreo, pueden generar datos sintéticos en el espacio de entrada.

Primeramente se resuelve el problema de inferencia **[38]** determinando las densidades de clase condicionales p (x | Ck) para cada clase Ck , siendo $X=(x_1,x_2,\ldots\ldots x_n)$ el vector que contiene las observaciones, de forma individual. También de forma separada se infiere la probabilidad de clase a priori p (Ck) . Luego, a partir del teorema de Bayes se encuentra la probabilidad posterior de clase p (Ck | x). En forma general, el denominador en el teorema de Bayes se especifica en términos de las cantidades que aparecen en el numerador.

La probabilidad conjunta p (x, Ck) se puede modelar directamente y luego se normaliza para obtener las probabilidades posteriores. Teniendo las probabilidades posteriores, se

utiliza la teoría de decisión para determinar la pertenencia a una clase de cada nueva entrada x.

2.4.1 Algoritmo EM

El método expectativa-maximización (EM) **[37]** es una clase de algoritmo bien conocido para la probabilidad máxima o estimación del máximo posteriori en los problemas con datos incompletos. En el caso de la clasificación semi-supervisada, los datos no etiquetados son considerados incompletos porque ellos no tienen las etiquetas de la clase. El EM básico asigna una primera probabilidad Dl al clasificador (por ejemplo, un clasificador Gaussiano) con los datos disponibles. Entonces, el clasificador se usa para asignar probabilísticamente peso a las etiquetadas de los ejemplos no etiquetados. Entonces un nuevo clasificador se entrena usando ambos datos, los originales etiquetados y los no etiquetados anteriormente, y el proceso se repite.

En el caso de los modelos de mezcla semi-supervisado, el algoritmo EM se puede considerar como la asignación de " etiquetas blandas" para los datos no etiquetados de acuerdo con el modelo actual **[41].**

Los diferentes resultados en la clasificación muestran que el método EM permite una explotación eficaz de los datos no etiquetados. Por ejemplo, Nigam **[38]** muestra que los datos no etiquetados usados en un documento con los métodos de EM en un problema de categorización puede reducir el error de clasificación por 30%.Por otro lado, Cohen recientemente mostró que el método EM sólo puede aumentar la exactitud de la clasificación si el modelo probabilístico asumido del clasificador empareja bien con la distribución de los datos.

La técnica de EM para el caso semi supervisado se puede aplicar con Naive Bayes **[42]** y se obtiene un algoritmo sencillo y atractivo. En primer lugar, un clasificador Naive Bayes se construye bajo la supervisión estándar de la cantidad limitada de etiqueta de los datos de entrenamiento. Luego, se lleva a cabo la clasificación de los datos no etiquetados con el modelo Naive Bayes, no toma nota de la clase más probable, pero si las probabilidades asociadas a cada clase. Se vuelve a generar un nuevo clasificador de Naive Bayes utilizando todos los datos con y sin etiqueta, utilizando las

probabilidades estimadas de la clase como la verdadera clase de etiquetas. Iterativamente se hace este proceso de clasificar los datos sin etiqueta y la reconstrucción del modelo de Naive Bayes hasta que converja a un clasificador estable y a un conjunto de etiquetas para los datos.

Si se tienen datos incompletos, se puede usar EM. La idea básica del algoritmo, es estimar la probabilidad de forma iterativa dados los datos que están presentes **[43]**.

Algoritmo (Expectativa de Maximización) véase **Anexo 1**.

El algoritmo EM puede ser utilizado para encontrar una (local) estimación de máxima verosimilitud (MLE) cuando están presentes datos no etiquetados. El algoritmo consiste en un paso de inicialización, donde a los parámetros del modelo se le asignan los valores iniciales, y dos pasos alternativos:

• Paso E: Los estadísticos se calculan bajo los parámetros del modelo actual (Es decir, teniendo en cuenta algunas asignaciones actuales de las instancias de las clases sin etiqueta)

• Paso M: los parámetros del modelo se actualizan para maximizar la similitud de los datos observados con estadísticos suficientes (es decir, actualización de la media y la varianza de las dos distribuciones de clase)

Lo más importante es que el algoritmo garantiza la verosimilitud de los datos.

2.4.2 Algoritmo Co-Training

Co-Training fue introducido por Blum y Mitchell (Blum y Mitchell, 1998) y está relacionado con el trabajo anterior del aprendizaje no supervisado de (Becker y Hinton, 1992). La idea es hacer uso de diferentes "puntos de vista" sobre los objetos para ser clasificados **[37].**

Co-Training **[40]** asume que las características pueden dividirse en dos grupos; Cada conjunto de características es suficiente para entrenar un buen clasificador; Los dos conjuntos son condicionalmente independientes dada la clase. Por primera vez dos clasificadores son entrenados por separado con los datos etiquetados. Cada

clasificador luego clasifica los datos no etiquetados, y el otro clasificador 'enseña' con los pocos ejemplos no etiquetados. Cada clasificador es re-entrenado con los ejemplos de entrenamiento adicional dados por el otro clasificador, y se repite el proceso.

Co-Training es similar a self-training con una diferencia crítica. En self-training un clasificador es usado para hacer predicciones en los datos no etiquetados, y entonces, estos datos se alimentan de nuevo en el algoritmo con etiquetas previstas. En Co-Training, dos clasificadores son utilizados, cada uno (potencial) operando en un punto de vista diferente de la misma instancia **[39]**.

Co-Training como self-training son un método Wrapper **[39]** que puede trabajar con dos clasificadores cualesquiera y ellos pueden asignar una puntuación de confianza para cada una de sus predicciones.y ellos pueden asignar una puntuación de confianza para cada una de sus predicciones (para decidir cuales etiquetas de instancias adicionar como datos de entrenamiento). Co-training es ampliamente aplicable a muchas tareas, especialmente cuando es posible obtener dos puntos de vista de cada caso.

Suponiendo que hay dos puntos de vista, el éxito del co-training depende de los siguientes dos supuestos.

Supuestos del Co-Training

1. Cada vista por sí sola es suficiente para hacer buenas clasificaciones, dado suficientes datos etiquetados.
2. Los dos puntos de vista son condicionalmente independientes dada la etiqueta de clase.

Co-Training puede ser entendido como inferencia bayesiana con información a priori condicional y se pueden aplicar técnicas estándares con el objetivo de mejorar la exactitud de la inferencia.

El algoritmo **[38]** puede ser visto como una variante de EM (secuencial).Este punto de vista permite generalizar Co-Training a lo largo de varias dimensiones, por ejemplo, teniendo en cuenta el ruido, mejores distribuciones anteriores, usar más lotes que el

entrenamiento en línea, sin saber que etiquetas son aplicadas en los puntos de prueba, etc.

2.4.3 Agrupamiento Semi-supervisado con el uso de Feedback

Este enfoque de la agrupación semi-supervisada **[44]** permite al usuario proporcionar retroalimentación iterativa a un algoritmo de agrupamiento. Los comentarios se incorporan en forma de restricciones, que el algoritmo de agrupación intenta satisfacer en futuras versiones. Estas restricciones permiten guiar al usuario hacia agrupamientos de los datos que el usuario encuentre más útil. El objetivo principal de este enfoque de agrupación semi-supervisado es permitir por ejemplo que un ser humano "dirija" el proceso de agrupamiento y pueda dividir las agrupaciones de un conjunto útil con el mínimo tiempo y esfuerzo. Un objetivo secundario de la agrupación semi-supervisada es dar al usuario una forma de interactuar y jugar con los datos que puedan entenderlo mejor. En el enfoque de la agrupación semi-supervisada se supone que el usuario humano tiene en su mente los criterios que les permitan evaluar la calidad de una agrupación. Esto no supone que el usuario es consciente de definir una agrupación bien, pero que, como con el arte, van a "lo saben cuando lo ven." Lo más importante es que, la agrupación semi-supervisada espera que un usuario escriba una función que defina el criterio de agrupamiento. En cambio, el usuario interactúa con el sistema de agrupamiento, que trata de aprender un criterio de que los rendimientos de las agrupaciones satisfagan al usuario. Como tal, uno de los principales desafíos de la agrupación semi-supervisada es encontrar maneras de obtener y hacer uso de los comentarios de los usuarios durante el agrupamiento.

La bondad de cualquier agrupación depende de lo bien que la métrica D coincide con el usuario (tal vez desconocida). Se propone que el usuario pueda imponer su modelo en la métrica a través del algoritmo de agrupamiento, haciendo que este (usuario) proporcione el algoritmo con comentarios, y lo que le permite alterar la métrica a fin de dar cabida a esos comentarios **[45]**.

2.4.4 Agrupamiento probabilístico semi-supervisado con restricciones

Es el problema de la partición de un conjunto de puntos de datos en un número determinado de grupos bajo supervisión limitada cuando se proporciona en forma de restricciones por parejas. Mientras que la agrupación es tradicionalmente considerada como una forma de aprendizaje no supervisado, ya que las etiquetas de clase no se dan, la inclusión de las limitaciones la hace una tarea de aprendizaje semi- supervisado, donde el rendimiento de algoritmos de agrupamiento sin supervisión puede mejorarse utilizando datos de entrenamiento etiquetados limitado.

Este tipo de restricción **[38]** se presenta normalmente como must-link y cannot-link en los puntos de datos: un must-link con restricción debe indicar que los dos puntos en la pareja deben ser colocados en el mismo grupo, mientras que un cannot-link con restricción lo contrario. Por otra parte el must-link y cannot-link a veces son llamados restricciones equivalentes y no equivalentes respectivamente. Por lo general, las restricciones son "débiles", es decir, restricciones que se violan, agrupamientos indeseables pero no prohibidos.

Los métodos propuestos para el agrupamiento semi-supervisado se dividen en dos categorías generales que se conocen como basada en restricciones y basada en la distancia. Los métodos basados en restricciones usan la supervisión para guiar el algoritmo a un particionamiento de datos que evita la violación de las restricciones. En los enfoques basados en la distancia, existe un algoritmo de agrupamiento que utiliza una función determinada de distancia entre los puntos de destino; sin embargo, la función de distancia con parámetros y los valores de los parámetros aprenden a colocar los puntos must-linked juntos y tomar los cannot-linked a parte **[45]**.

También existe un enfoque de agrupación semi-supervisada basada en los HMRFs [11]que combina los métodos basados en la restricción y los basados en la distancia en un enfoque de un modelo probabilístico unificado. La formulación probabilística conduce a la agrupación de una función objetivo derivada de la probabilidad conjunta de los

[11] Campos de modelos ocultos de Markov

puntos de datos observados, las asignaciones de agrupamiento, y los parámetros del modelo generativo.

Esta función objetivo puede ser optimizada con EM **[38]**, algoritmo de estilo agrupación, HMRF-KMeans. HMRF-KMeans puede ser utilizado para realizar la agrupación semi-supervisada con una amplia clase de funciones de distorsión (a distancia), conocidas como, divergencias Bregman (Banerjee et al., 2005), que incluyen una gran variedad de distancias útiles, por ejemplo, la divergencia de KL, la distancia euclidiana al cuadrado, la divergencia I, y la distancia Itakuro- Saito.

La última interpretación parece estar más en línea con la mayoría de las aplicaciones, donde el objetivo es el mismo que en el aprendizaje supervisado **[38]**: predecir un valor objetivo de un X_i dado. Sin embargo, esta visión no se aplica fácilmente si el número y la naturaleza de las clases no son conocidas de antemano, sino que se desprende de los datos.

2.4.4.1 MPCKMeans

MPCKMeans de Bilenko, Basu, y Money (2004).Este algoritmo **[46]** combina el agrupamiento KMeans con restricciones con una métrica de aprendizaje para minimizar una sola función objetivo. MPCKMeans utiliza restricciones débiles a pares entre ambas instancias para situar los centroides iniciales de los cluster y para influir en la agrupación a través de la función objetivo.

Bilenko et al. **[46]** representa la distancia euclidiana con una métrica de una matriz A simétrica definida positivamente. A es una métrica Mahalanobis. Donde, X es el conjunto de instancias de datos; M y C son los conjuntos de restricciones must-link y cannot-link, respectivamente; μh representa el centroide del cluster h, donde $h \in \{1,. . ., K\}$, y la matriz Ah representa la métrica para el cluster h. MPCKMeans genera un particionamiento K del conjunto de datos X que (localmente) minimiza la función objetivo (Bilenko, Basu, y Mooney, 2004):

$$J_{MPCKmeans} = \sum_{x_i \epsilon X} \left(\|x_i - \mu_{x_i}\|^2_{A_{x_i}} - \log(\det(A_{x_i})) \right)$$

(2.1) $$+ \sum_{(x_i, x_j, w) \epsilon M} w f M(x_i, x_j) \P \left[\mu_{x_i} \neq \mu_{x_j} \right]$$

$$+ \sum_{(x_i, x_j, \bar{w}) \epsilon M} \bar{w} f_c(x_i, x_j) \P \left[\mu_{x_i} = \mu_{x_j} \right]$$

(2.2) $$f_M(x_i, x_j) = \frac{1}{2} \|x_i - x_j\|^2_{A_{x_i}} + \frac{1}{2} \|x_i - x_j\|^2_{A_{x_j}}$$

(2.3) $$f_c(x_i, x_j) = \|x - x\|^2_{A_{x_i}} - \|x - x\|^2_{A_{x_i}}$$

Esta ecuación es la misma que la utilizada por Bilenko et al.con una pequeña modificación: El primer término de JMPCKmeans es un intento de maximizar la función de logaritmo de verosimilitud del agrupamiento KMeans. El segundo y tercer término incorpora los costos de violar las restricciones en M y C.

El algoritmo MPCKMeans utiliza expectativa-maximización (EM) para generar el cluster de X que minimiza localmente. El paso E consiste en asignar cada punto a un cluster que minimiza JMPCKmeans desde la perspectiva de los puntos de datos, dadas la asignaciones previas de los puntos a los clusters. El paso M consiste en dos partes: re-estimar los centroides del cluster dado las asignaciones del cluster en el paso E, y la actualización de las matrices métricas para disminuir JMPCKmeans.

2.5 Separación de baja densidad

El enfoque de separación de baja densidad **[38]** plantea que: si dos puntos X_1, X_2 en una región de alta densidad están cerca, entonces también lo deberían estar los resultados correspondientes Y_1, Y_2. Teniendo en cuenta que por transitividad, esta suposición implica que si dos puntos están unidos por un camino de alta densidad (por ejemplo, si pertenecen al mismo grupo), entonces sus salidas es probable que estén

cerca. Si, en cambio, están separados por una región de baja densidad, sus resultados no deben estar cerca. Teniendo en cuenta que el supuesto semi-supervisado se aplica tanto a la regresión como a la clasificación.

Separación de baja densidad: El límite de la decisión debe recaer en una región de baja densidad.

La equivalencia es fácil de ver: Un límite de decisión en una región de alta densidad cortaría un agrupamiento en dos clases diferentes, muchos objetos de diferentes clases en el mismo grupo requieren ser cortados por el límite de decisión, es decir, para ir a través de una región de alta densidad.

2.5.1 Soporte de Máquinas de Vectores Transductivo

En contraste con el aprendizaje de una regla de predicción general, V. Vapnik **[38]** propuso la creación del aprendizaje transductivo donde las predicciones se hacen sólo en un número fijo de puntos de prueba conocida. Esto permite que el algoritmo de aprendizaje aproveche la ubicación de los puntos de prueba, por lo que es un tipo particular de problemas de aprendizaje semi-supervisado. Soporte de Máquinas de Vectores Transductivos (TSVMs) pone en práctica la idea del aprendizaje transductivo mediante la inclusión de puntos de prueba en el cálculo del margen. Como el objetivo es aprender una regla que clasifique con precisión todos los ejemplos que permanecen en la base de datos de acuerdo a su relevancia. Es evidente que este problema se puede considerar como un problema de aprendizaje supervisado. Pero es diferente de muchos otros (inductivo) problemas de aprendizaje en al menos dos aspectos.

En primer lugar, el algoritmo de aprendizaje no necesariamente tiene que aprender una regla general, pero sólo se necesita predecir con precisión para un número finito de ejemplos de prueba (es decir, los documentos en la base de datos). En segundo lugar, los ejemplos de prueba se conocen a priori y se puede observar por el algoritmo de aprendizaje durante el entrenamiento. Esto permite que el algoritmo de aprendizaje explote cualquier información que pudiera estar contenida en la ubicación de los ejemplos de prueba. El aprendizaje transductivo es un caso particular del aprendizaje

semi-supervisado, ya que permite que el algoritmo de aprendizaje aproveche los ejemplos sin marcar en el equipo de prueba **[39]**.

2.6 Métodos basados en grafos

Durante los últimos pares de años, el área más activa en la investigación del aprendizaje semi supervisado han sido los métodos basados en grafos **[38]**. El denominador común en estos métodos es que los datos son representados por nodos de un grafo, las aristas son etiquetadas con la distancia a pares (y la arista faltante corresponde con la distancia infinita).

Por lo general, consiste en la predicción de las etiquetas de los nodos no etiquetados. Las etiquetas conocidas se utilizan para difundir información a través de los grafos con el fin de etiquetar todos los nodos. Por esta razón, este tipo de algoritmo es intrínsecamente transductivo, es decir, que sólo devuelve el valor de la función de la decisión sobre los puntos sin marcar y no la decisión de la función en sí. Sin embargo, ha sido un trabajo reciente producir soluciones inductivas con el fin de ampliar los métodos basados en grafos.

La propagación de la información en grafos puede servir para mejorar la clasificación teniendo en cuenta los datos no etiquetados.

Los métodos basados en grafos son transductivos por naturaleza. Los grafos en el aprendizaje semi-supervisado desempeñan un papel importante. En la mayoría de los casos, el aprendizaje semi-supervisado basado en grafo comienza por la construcción de un grafo de los datos de entrenamiento. Teniendo en cuenta los datos de entrenamiento, los vértices son las instancias etiquetadas y no etiquetadas.

Claramente, si u es un grafo grande, el tamaño de los datos sin etiqueta, es grande. Una vez que el grafo es construido, el aprendizaje implica la asignación de valores a y de los vértices en el grafo. Esto es posible gracias a las aristas que conectan vértices etiquetados a vértices no etiquetados. Las aristas de los grafos son por lo general no dirigidas. Una arista entre dos vértices xi, xj tradicionalmente representa la similitud de

las dos instancias. Wij será el peso de la arista. La idea es que si wij es grande, entonces las dos etiquetas yi, yj se espera que sean la misma.

Antes de zambullirse en algoritmos específicos, se puede imaginar como un proceso que se extiende o se propaga etiquetas de los vértices etiquetados **[39]** para los vértices no etiquetados a través de las aristas del grafo, con la cantidad de propagación en función de los pesos de las aristas. Donde los vértices intermedios sin etiqueta se pueden utilizar como escalones, lo que permite a una etiqueta difundirse a las instancias sin etiqueta que no están directamente conectados a cualquier vértice etiquetado. Por lo tanto, los pesos en las aristas del grafo son de gran importancia.

Los problemas del aprendizaje semi-supervisado basados en grafo pueden formularse como un grafo de corte (Blum y Chawla, 2001, Blum et al, 2004) **[39]**,aquí, las aristas positivas etiquetadas son tratadas como vértices " fuente", como si un líquido fluyera fuera de ellos y por los bordes. Del mismo modo, las aristas negativas son etiquetadas de vértices "sumideros", donde el líquido desaparece. El objetivo es encontrar un conjunto mínimo de aristas que elimine todos los bloques desde el flujo de nacimiento hasta los sumideros. Esto define un "Corte", o una partición del grafo en dos conjuntos de vértices. El "tamaño de corte" se mide por la suma de los pesos en los bordes que definen el corte. Una vez que el grafo se divide, los vértices de conexión a las fuentes están etiquetados como positivos, y aquellos a los sumideros están etiquetados como negativos.

2.6.1 Propagación de etiquetas

Muchos algoritmos de aprendizaje semi-supervisados **[39]** se basan en la geometría de los datos inducidos por ejemplos etiquetados y sin etiquetar, para mejorar los métodos supervisados que sólo usan los datos etiquetados. Esta geometría puede ser, naturalmente, representado por un grafo empírico que represente los datos de entrenamiento y las aristas representen similitudes entre ellos. Estas similitudes están dadas por una matriz W: W_{ij} no es cero si X_i y X_j son vecinos.

Dado el grafo g, una idea simple para el aprendizaje semi-supervisado es propagar las etiquetas en el grafo. A partir de los nodos 1, 2,. . ., L etiquetados con un valor conocido

(1 o -1) y los nodos l+1,. . ., N con etiqueta 0, cada nodo empieza a propagar su etiqueta a sus vecinos, y el proceso se repite hasta la convergencia.

Propagación de Etiquetas[40]: El algoritmo de propagación de etiquetas original fue propuesto por (Zhu y Ghahramani, 2002a). En este algoritmo se formula el problema como una forma de propagación en un grafo, donde los nodos etiquetados se propagan a los nodos vecinos de acuerdo a su proximidad. En este proceso se fijan las etiquetas de los datos etiquetados.

Las etiquetas se propagan a través de las aristas. Mientras mayor es el peso de las aristas mejor se propagan las etiquetas. Se define una matriz de transición probabilística P de n × n. Donde P_{ij} es la probabilidad de transición del nodo i al j. También se define una matriz YL de etiquetas de l × C, las i filas son un vector indicador para $y_i, i \in L: Y_{ic} = \delta(y_i, c)$ Se calculan algunas etiquetas f para los nodos. f es una matriz n × C, las filas pueden ser interpretadas como la probabilidad de distribución de las otras etiquetas. La inicialización de f no es importante.

El algoritmo de propagación de etiqueta es el siguiente **[40]**:

1. Propagar $f \leftarrow Pf$
2. Ajustar los datos etiquetados fL = YL.
3. Se repite desde el paso 1 hasta que f converge.

Otro algoritmo de propagación de etiqueta fue propuesto por Zhou et al. (2004) **Label spreading [38]:** en cada paso de un nodo i recibe una contribución de sus vecinos j (ponderado por el peso normalizado de la arista (i, j)), y una pequeña contribución adicional dado por su valor inicial.

2.6.1.1 Markov Random Walks

Los paseos aleatorios de Markov es un algoritmo diferente **[38],** basado en la propagación de etiqueta de un grafo de similitud propuesto por primera vez por Szummer y Jaakkola (2002b).MRW es considerado sobre un grafo con probabilidades de transición de i a j,

$$(2.4) \qquad p_{ij} = \frac{W_{ij}}{\sum_k W_{ik}}$$

con el fin de estimar las probabilidades de las etiquetas de la clase. W_{ij} está dada por un núcleo gaussiano de vecinos y 0 para los no vecinos, y $W_{ii} = 1$. Cada punto de datos X_i está asociado con una probabilidad P (y = 1 | i) de ser de la clase 1. Dado un punto X_k, podemos calcular la probabilidad P (t) ($y_{start} = 1 \mid k$) que comienza de un punto de la clase y start = 1, dado a que arriba a x_k después de t pasos de las calles aleatorias.

El rendimiento de este algoritmo depende fundamentalmente del hiperparámetro t (la longitud de MRW). Este parámetro es elegido por la validación cruzada (si se dispone de suficientes datos) o por la heurística (que corresponde intuitivamente que la cantidad de propagación que permiten en el gráfico, es decir, a la escala de los grupos que interesa).

Una alternativa en el uso de los paseos aleatorios en el grafo es asignar a punto X_i una etiqueta en dependencia de la probabilidad de arribo de un ejemplo etiquetado positivamente cuando se realiza los paseos aleatorios a partir de Xi hasta que un ejemplo etiquetado se encuentra.

Cuando X_i es un ejemplo etiquetado, $P(y_{end} = 1|i) = \delta_{yi}1$, y cuando es no etiquetado la relación es :

$$(2.5) \qquad P(yend = 1 \mid i) = \sum_{j=1}^{n} P(yend = 1 \mid j)pij,$$

2.6.1.2 Combinación de múltiples grafos

Cuando múltiples grafos **[38]** están disponibles, es natural para incorporarlas como fuentes de información adicionales. Por ejemplo, las proteínas pueden ser representadas en forma de grafos de acuerdo con sus secuencias de aminoácidos, las

estructuras, las interacciones, o de otro tipo de relaciones. Seleccionado un grafo de m grafos sería relativamente fácil. Se puede resolver el problema de aprendizaje con cada grafo, y sólo tiene que seleccionar el mejor en términos de, por ejemplo, el error de validación cruzada.

2.7 Aplicaciones en redes de predicción de funciones de proteínas

En la biología computacional, es común representar el dominio de conocimiento mediante grafos. Con frecuencia existen múltiples grafos para el mismo conjunto de nodos, que representan información de diferentes fuentes, y un solo grafo no es suficiente para predecir etiquetas de clase de nodos no etiquetados de forma fiable. Una forma de mejorar la fiabilidad es con la integración de grafos múltiples, ya que los grafos individuales son en parte independientes y complementarios.

El método de grafos combinados se aplica a la predicción funcional de clases de las proteínas de la levadura. Cuando es comparado con grafos individuales, el grafo combinado con los pesos optimizados mejora significativamente los resultados que cualquier grafo único. Cuando comparamos con la programación semi-definida basada en máquinas de vectores de soporte (SDP / SVM), muestra una precisión notable en un corto tiempo **[38]**.

En la biología computacional, muchos tipos de datos genómicos son con frecuencia representados utilizando grafos. Los nodos corresponden a los genes o las proteínas, y las aristas se corresponden con las relaciones biológicas entre ellos. Algunas proteínas tienen clases de funciones conocidas a través de experimentos biológicos, mientras que otras son desconocidos debido al elevado costo y esfuerzo que exige. Es útil para predecir la función desconocida de las proteínas a fin de orientar y ayudar a entender en los experimentos los mecanismos complejos de la célula.

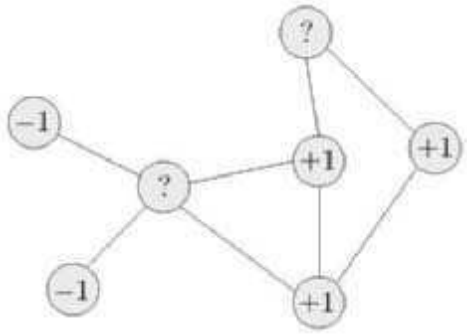

Ilustración 3. Grafo para la predicción de funciones de proteínas

El problema de la predicción de funciones puede ser mostrado como un grafo indirecto. Enfocándose en una clase funcional particular, al final el problema es de clasificación de dos clases. Una proteína que la etiqueta de clase es conocida es anotada +1y sino -1.La etiqueta positiva indica que la proteína pertenece a la clase de otra manera es que no. Cuando un grafo está definido el problema puede ser resuelto dentro de un framework[12] de aprendizaje semi-supervisado basado en grafos gracias a los recientes progresos de (Zhou et al., 2004; Belkin and Niyogi, 2003b; Zhu et al., 2003b; Chapelle et al,2003)

Por lo general, múltiples grafos están disponibles para representar el mismo conjunto de proteínas en términos de varias fuentes de información. Por ejemplo, un conjunto de aristas pueden representar interacciones físicas de las proteínas, relaciones de regulación de genes, la cercanía de una vía metabólica y similitudes entre las secuencias de proteínas.

Cada fuente contiene información parcialmente independiente y complementaria acerca de la tarea en cuestión. Sin embargo, ninguna fuente única de información es suficiente para identificar las funciones de la proteína de forma fiable. Una forma de mejorar la fiabilidad es la integración de múltiples fuentes. En la biología computacional, una serie de métodos han sido propuestos para la clasificación de proteínas basados en redes tales como mayoría de votos, métodos basados en grafos (al Schwikowski et al.,

[12] conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular

2000; Hishigaki et al, 2001), métodos Bayesiano (Vázquez et al, 2003), métodos de aprendizaje discriminativo (Vert y Kanehisa, 2003;Lanckriet et al, 2004c), y la integración probabilística mediante las puntuaciones del log-verosimilitud.

2.8 Conclusiones

El aprendizaje semi-supervisado tiene un valor práctico enorme. En muchas tareas, existe escasez de datos etiquetados. Las etiquetas pueden ser difíciles de obtener porque requieren expertos humanos, dispositivos especiales o experimentos costosos y lentos. El SSL es atractivo porque puede utilizar potencialmente tanto los datos etiquetados como sin etiquetar para lograr un mejor rendimiento que el aprendizaje supervisado. Desde una perspectiva diferente, este tipo de aprendizaje puede alcanzar el mismo nivel de rendimiento que el aprendizaje supervisado, pero con menor número de casos etiquetados. Esto reduce el esfuerzo de anotación, lo que conduce a un costo reducido.

La predicción de estructura de proteínas puede tomar meses de trabajo de laboratorio a expertos para identificar la estructura 3D de una sola proteína. Por esta razón usualmente se emplea la representación mediante cadenas de aminoácidos para su comparación, esta representación es la utilizada en este trabajo.

En el estudio realizado se identificaron tres grandes grupos de métodos de clasificación semi-supervisada: los Modelos Generativos, Separación de Baja Densidad y los métodos Basados en Grafos.

Capítulo 3 . Diseño del experimento y análisis de los resultados.

En el presente capítulo se realiza el diseño del experimento donde se explica el método utilizado para modificar la base de casos con la que se cuenta, luego se aplican los algoritmos de los diferentes enfoques de aprendizaje, y se analizan los resultados ofrecidos por dichos algoritmos escogidos. También se justifican las herramientas utilizadas para dicha experimentación.

3.1 Descripción de las herramientas empleadas

3.1.1 Matlab

Es un software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows y Apple Mac OS X.

Matlab® integra computación, visualización, y programación, en un entorno fácil de usar donde los problemas y las soluciones son expresados en la más familiar notación matemática. Los usos más familiares son:

- Matemática y Computación
- Desarrollo de algoritmos
- Análisis de datos, exploración y visualización
- Graficas científicas e ingenieriles
- Desarrollo de aplicaciones, incluyendo construcción de interfaces gráficas de usuario

Es un sistema interactivo cuyo elemento básico de almacenamiento de información es la matriz, que tiene una característica fundamental y es que no necesita dimensionamiento. Esto le permite resolver varios problemas de computación técnica (especialmente aquellos que tienen formulaciones matriciales y vectoriales) en una fracción de tiempo similar al que se gastaría cuando se escribe un programa en un lenguaje no interactivo como C o FORTRAN

Matlab se ha desarrollado sobre un período de años con entradas provenientes de muchos usuarios, en los entornos universitarios. Es la herramienta instructiva estándar para cursos avanzados e introductorios en matemáticas, ingeniería y ciencia. En la industria es la herramienta escogida para investigación de alta productividad, desarrollo y análisis.

Presenta una familia de soluciones a aplicaciones específicas de acoplamiento rápido llamadas ToolBoxes. Los ToolBoxes son colecciones muy comprensibles de funciones, o archivos (M-files) que extienden el entorno de Matlab para resolver clases particulares de problemas, Algunas áreas en las cuales existen ToolBoxes disponibles son:

- Procesamiento de señales
- Sistemas de control
- Redes neuronales
- Lógica difusa
- Wavelets
- Simulación
- Bioinformática

Por las razones expuestas anteriormente se ha decidido que el uso de Matlab es apropiado para el experimento correspondiente a esta investigación.[4] En este caso se ha utilizado su versión 7.0 **[47]**.

3.1.2 Weka

Weka es un software conocido para aprendizaje automático y minería de datos escrito en Java y desarrollado en la Universidad de Waikato. Es un programa libre, distribuido bajo licencia GNU-GPL. Contiene una colección de herramientas de visualización y algoritmos para análisis de datos y modelado predictivo, unidos a una interfaz gráfica de usuario para acceder fácilmente a sus funcionalidades. La versión original fue una aplicación para modelar algoritmos implementados en otros lenguajes de programación,

más unas utilidades para preprocesamiento de datos desarrolladas en C para hacer experimentos de aprendizaje automático. Esta versión original se diseñó inicialmente como herramienta para analizar datos procedentes del dominio de la agricultura, pero la versión más reciente basada en Java (Weka 3), que empezó a desarrollarse en 1997, se utiliza en muchas y muy diferentes áreas, en particular con finalidades docentes y de investigación **[21].** Para éste estudio se ha utilizado la versión 3.4.12.

3.2 Descripción de la estructura de la información de las especies

Los organismos empleados en la experimentación son dos tipos de Chlamydia, la Chlamydia trachomatis y la Chlamydophila pneumoniae Estos organismos son bacterias muy comunes causantes de diferentes enfermedades en los seres humanos. Los genomas completos se pueden descargar de la base de datos GenBank.

El género Chlamydia incluye tres especies: C. trachomatis, C. muridarum y C. suis.

Chlamydia es un grupo de bacterias de tamaño pequeño, (en relación a otras bacterias) y de forma esferoidal. Su principal característica es el ciclo replicativo intracelular, lo cual las convierte en parásitos obligados.

Chlamydia trachomatis, causa tracoma y ceguera, infecciones oculares, genitales y neumonías. Algunos individuos desarrollan el Síndrome de Reiter, que no tiene cura.

C. pneumoniae el 90% cursa con infección subclínica. Pudiendo producir faringitis, otitis media, neumonía atípica. Se cree contribuye en la formación de las placas ateroescleróticas

La Chlamydia trachomatis y la C. pneumoniae son especies cercanas, su relación en la evolución se muestra en el siguiente árbol filogenético.

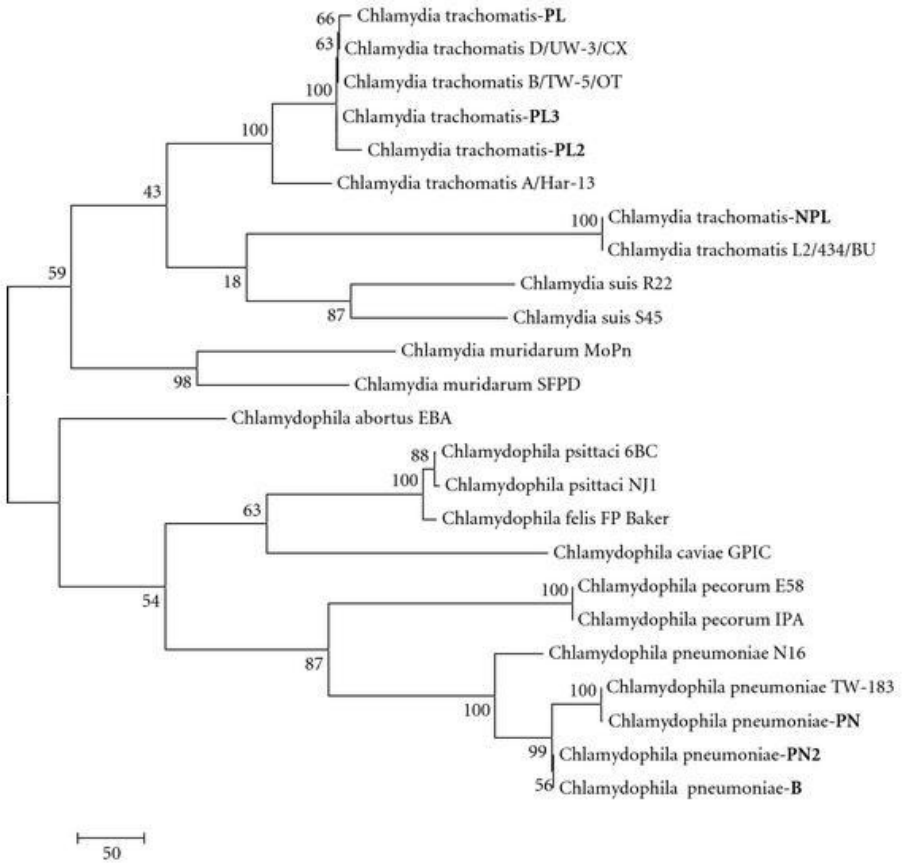

Ilustración 4. Árbol filogenético de la Chlamydia

Esta cercanía permitió su selección para realizar los experimentos que aquí se exponen. La trachomatis posee un genoma circular de longitud sobre el millón bp. Estas secuencias son largas por lo que puede tomar mucho tiempo descargar completamente los genomas.

La forma en la que se almacena la información es la siguiente:

seqtrachomatis =

1) LocusName: 'NC_000117'
2) LocusSequenceLength: '1042519'
3) LocusTopology: 'circular'

4) LocusMoleculeType: 'DNA'

5) LocusGenBankDivision: 'BCT'

6) LocusModificationDate: '04-DEC-2007'

7) Definition: [1x49 char]

8) Accession: 'NC_000117'

9) Version: 'NC_000117.1'

10) Project: 'GenomeProject:45'

11) Keywords: []

12) Segment: []

13) Source: 'Chlamydia trachomatis D/UW-3/CX'

14) SourceOrganism: [2x61 char]

15) Reference: {[1x1 struct] [1x1 struct] [1x1 struct]}

16) Comment: [3x65 char]

17) Features: [18685x75 char]

18) **CDS:** [1x895 struct]

19) Sequence: [1x1042519 char]

20) SearchURL: [1x70 char]

Donde cada uno de los elementos almacena la siguiente información:

1. Nombre del locus
2. Longitud de la secuencia del locus
3. Topología del locus
4. Tipo de molécula del locus
5. División del locus en el GenBank

6. Fecha de modificación del locus
7. Definición
8. Número de acceso
9. Versión
10. Proyecto
11. Palabras Claves
12. Segmento
13. Fuente
14. Organismo fuente
15. Referencia
16. Comentario
17. Características
18. CDS
19. Secuencia
20. Dirección en la Web

Para comparar los dos genomas se sigue el enfoque del alineamiento a pares entre las secuencias de ambos genomas. El enfoque sería comparar todos los ORF en los dos genomas. Sin embargo los datos de GenBank incluyen más información sobre los genes en la secuencia. Esta información útil es almacenada en el campo **CDS** de la estructura anterior. Chlamydia trachomatis tiene 895 regiones codificantes mientras que Chlamydophila pneumoniae tiene 1112.

La mayoría de los CDS contienen en **translation** la secuencia de aminoácidos. El primer CDS por ejemplo en la Chlamydia trachomatis es una proteína hipotética de longitud de 591 residuos. Aunque se debe tener en cuenta que algunos CDS pueden estar vacíos y para establecer la comparación primero deberían rellenarse.

A continuación se muestra la estructura de un CDS

ans =

1) location: 'complement(73364..73477)'
2) gene: []
3) product: 'hypothetical protein'
4) codon_start: '1'
5) indices: [73477 73364]
6) protein_id: 'NP_444613.1'
7) db_xref: 'GeneID:963699'
8) note: [2x45 char]
9) **translation**: 'MLTDQRKHIQMLHKHNSIEIFLSNMVVEVKLFFKTLK*'
10) text: [11x52 char]

Donde cada uno de los elementos almacena la siguiente información:

1. Localización
2. Gen
3. Producto
4. Codón de inicio
5. Índices
6. Id de la proteína
7. Referencia en la base de datos
8. Nota
9. Translación
10. Texto

3.3 Detección de genes ortólogos usando el algoritmo BBH

Matlab dispone de un ToolBox para bioinformática el cual cuenta con funciones para el alinemiento tales como nwalign y swalign entre otras en dependencia del alineamiento que se desee realizar.

El alineamiento de un gen de la Chlamydia trachomatis con todos los genes de Chlamydophila pneumoniae tarda alrededor de un minuto en una computadora personal Intel® Pentium 4, 2.0 GHz con sistema operativo Windows® XP. El cálculo del alineamiento para los 895 CDS de la Chlamydia trachomatis toma alrededor de 12 horas en la misma computadora.

La distribución de la puntuación que reporta el algoritmo es afectado por la longitud de las secuencias, con secuencias largas se reportarán puntajes muchos más altos que para secuencias cortas. Para observar el comportamiento de los puntajes en alguna gráfica estos se pueden normalizar por diferentes vías. Una vía simple sería dividir por la longitud de la secuencia.

Otro factor importante además del tiempo de duración del alineamiento es la memoria teniendo en cuenta la dimensión de la matriz de puntaje que se genera. En estos casos las matrices Sparse constituyen una manera interesante de almacenar los datos de una forma más eficiente.

Los puntajes positivos indican un buen alineamiento y si la mayoría de estos son negativos los datos no son útiles para este tipo de estudio.

Los ceros en la matriz de puntajes indican que un determinado CDS en un organismo no tiene coincidencia con otro CDS del otro organismo.

3.3.1 Buscando genes ortólogos

La homología es un concepto central: la homología es la relación que existe entre dos partes orgánicas diferentes cuando sus determinantes genéticos tienen el mismo origen evolutivo. La esencia del análisis de secuencia es la detección de relaciones homólogas mediante búsquedas en bases de datos de secuencia.

Ortología (especiación)	Paralogía (duplicación)
Misma función	Funciones diferentes
Especies diferentes	Relaciones en un mismo organismo

Tabla 1. Tipos de Homología de Secuencia

La manera más obvia de saber si dos genes son ortólogos es si tienen un alto puntaje con respecto a otro gen en la matriz. Si el puntaje significativo es BBH y es muy probable que sean ortólogos.

Los ortólogos que reporta el algoritmo BBH con estas secuencias y un puntaje mayor que 0.9 utilizando Matlab pueden observarse en el **Anexo 2**.

3.4 Co- agrupamiento Espectral

Un agrupamiento de grafo espectral utiliza los valores propios de la matriz de adyacencia de los datos para asignar el mapa original inherente a las relaciones de co-ocurrencia en un nuevo espacio espectral. Después de la proyección, las secuencias son simultáneamente divididas en grupos con reducción mínima de optimización. **[49]**

De acuerdo con la teoría de grafos espectrales se escoge k vectores singulares izquierdos y k vectores singulares derechos de la matriz reformada

$$\mathrm{N}A = D_s^{-\frac{1}{2}} A D_p^{-\frac{1}{2}} \qquad (2.6)$$

presentando una mejor aproximación de los vectores de las filas y las columnas en el nuevo espacio espectral. D_s y D_p son matrices diagonales de cada una de las secuencias, y se definen como:

$$(2.7) \quad D_s(i,i) = \sum_{j=1}^{n} a_{ij}\ , i = 1, \dots, m, D_p(j,j) = \sum_{i=1}^{m} a_{ij}\ , j = 1, \dots, n$$

Ls denota una matriz m x k de k vectores singulares izquierdos. El objetivo es llevar a cabo un subconjunto de doble agrupamiento, se crea una nueva matriz (m + n) × k PV para reflejar la proyección de los vectores fila y columna en el nuevo espacio espectral en una menor dimensión como:

$$(2.8) \quad \text{PV} = \begin{matrix} D_s^{-1/2} & L_s \\ D_p^{-1/2} & R_p \end{matrix}$$

Todos estos pasos del algoritmo son resumidos de la siguiente forma:

Algoritmo: Algoritmo Co-agrupamiento Espectral

Entrada: Secuencia del genoma de especie Chlamydia trachomatis S y secuencia del genoma de especie C pneumoniae P.

Salida: Un conjunto C = {C 1, • • • Ck} de subconjuntos de k ortólogos.

1. Construir una matriz A con el uso de los puntajes de similitud, cuyo elemento es determinado por el alineamiento previamente realizado.

2. Calcular las dos matrices diagonales D_s y D_p de A;

3. Formar una nueva matriz NA = $D_s^{-\frac{1}{2}} A D_p^{-\frac{1}{2}}$;

4. Mejorar la operación de los SVD[13](valores singulares de descomposición) de NA, y obtener los k vectores singulares izquierdos y los derechos Ls y Rp, y combinar la

[13] Sigla del inglés Singular Values Descomposition

transformación de los vectores fila y columna para crear un nueva matriz de proyección PV (vectores de proyección);

5. Ejecutar un algoritmo de agrupamiento en PV y devolver los subconjuntos de Co-cluster de S y P, $C_j = (S_j,P_j)$.**[49]**

3.5 Selección de los algoritmos de agrupamiento

Luego de aplicar el Co-agrupamiento Espectral hasta el paso cuatro falta elegir el algoritmo de agrupamiento del paso cinco para ejecutarlo con las secuencias genómicas seleccionadas. Se propone realizar experimentos con algoritmos de agrupamiento no supervisado Single-link, Average link, Complete-link, KMeans y comparar los resultados con el algoritmo semi-supervisado MPCKMeans tomando como punto de partida los resultados que arroja el algoritmo BBH.

3.5.1 Algoritmos Single-link, Average link, Complete-link, KMeans

Los algoritmos detallados fueron los no supervisados seleccionados para el paso cuatro del Co-agrupamiento Espectral. A continuación se detallan las características de cada uno de ellos. El algoritmo semi-supervisado seleccionado fue el MPCKMeans el cual se detalló en el capítulo anterior.

3.5.1.1 KMeans

KMeans es un algoritmo de agrupamiento particional que lleva a cabo la reubicación iterativa para particionar un conjunto de datos en k grupos y el conjunto de los medios (también conocido como centroides)**[50].** Este algoritmo de agrupamiento se utiliza ampliamente debido a su implementación sencilla. El algoritmo toma el número de clusters (k) como entrada, el número de grupos suele ser elegido por el usuario. El procedimiento para la agrupación de KMeans es el siguiente:

1. En primer lugar, el usuario trata de estimar el número de clusters.
2. Se elige aleatoriamente N puntos en los k grupos.
3. Calcular el centroide de cada cluster.
4. Cada punto, se mueve al cluster más cercano.

5. Repitir los pasos 3 y 4 hasta que no haya más puntos que se mueven a otros cluster.

El algoritmo KMeans es uno de los algoritmos de agrupamiento más rápidos y sencillos. Sin embargo, tiene un gran inconveniente. Los resultados del algoritmo de KMeans pueden cambiar en corridas sucesivas, porque los grupos iniciales se eligen al azar. Como resultado de ello, el investigador tiene que evaluar la calidad de la agrupación obtenida**[51]**.

Para observar el pseudocódigo algoritmo KMeans véase **Anexo 3**.

3.5.1.2 Single-link

El agrupamiento single-link o single-link clustering, solo calcula la distancia entre clusters como la distancia entre los vecinos más cercanos. Se mide la distancia entre cada miembro de un cluster con cada miembro de el otro cluster, y toma el mínimo de estos. El criterio de combinación es local. Prestan una atención exclusiva al área donde son más cercanos los dos clusters el uno al otro. Otras partes, más distantes de la agrupación y la estructura de las agrupaciones en general no se toman en cuenta **[52] .**

3.5.1.3 Complete-link

El agrupamiento complete-link o complete-linkage, calcula la distancia entre los vecinos más lejanos. Se necesita el máximo de distancia medidas entre todos los miembros de un cluster a cada miembro del otro cluster. Esto es equivalente a la selección de los pares de agrupamientos cuya combinación tiene el diámetro más pequeño. En el complete-linkage el criterio de fusión no es local; toda la estructura de la agrupación puede influir en las decisiones de la fusión **[52] .**

Tanto la agrupación single-link como complete-link tienen interpretaciones teóricas de grafo.

3.5.1.4 Average link

El agrupamiento average link mide la distancia promedio entre cada miembro de un cluster a cada miembro del otro cluster. Une en cada iteración el par de grupos con

mayor cohesión y actualiza los centroides de los agrupamientos gammas y cohesiones de la posibles fusiones de los agrupamientos recién creados con los grupos restantes**[53]**

3.6 Resultados de los experimentos del Co-agrupamiento Espectral con los algoritmos no supervisados

Para observar los resultados de la aplicación del Co-agrupamiento Espectral con los algoritmos **Single Link y Complete Link** véase **Anexo 4**.

Para observar los resultados de la aplicación del Co-agrupamiento Espectral con el algoritmo **Average Link** véase **Anexo 5**.

Para observar los resultados de la aplicación del Co-agrupamiento Espectral con el algoritmo **KMeans** véase **Anexo 6**.

3.7 Resultados del experimento del Co-agrupamiento Espectral con el algoritmo MPCKMeans

Con la matriz de puntuación se puede observar la distribución del alineamiento de genes de las especies Chlamydophila pneumoniae con la Chlamydia trachomatis.

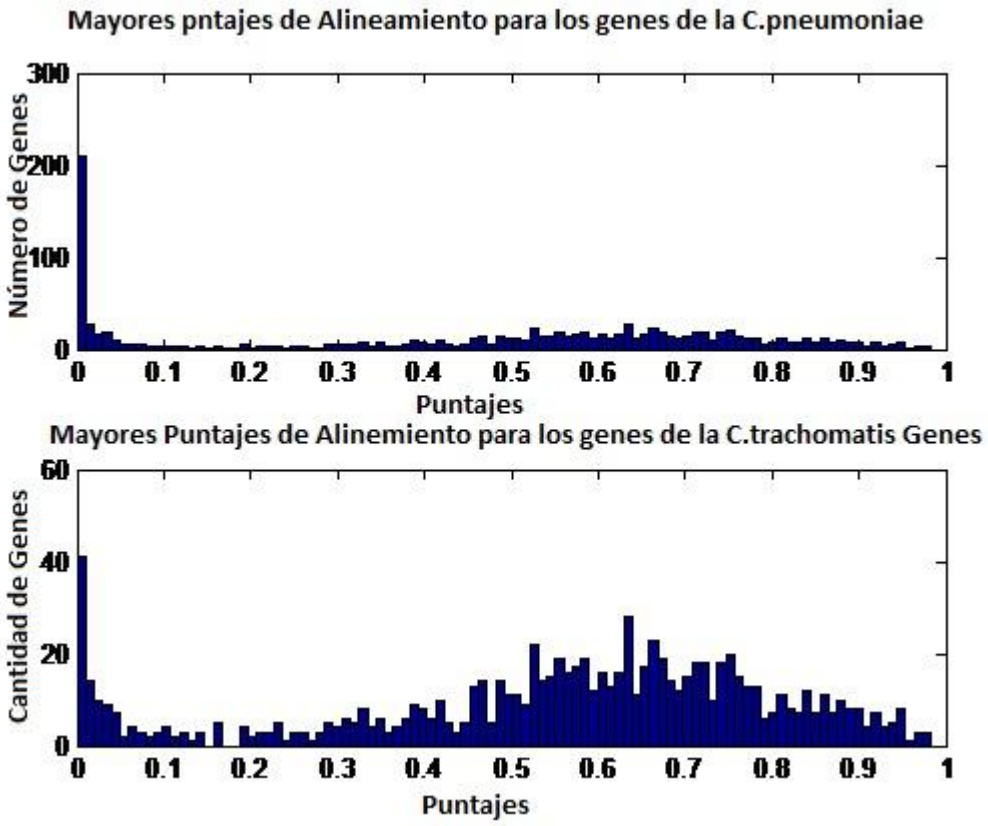

Ilustración 5. Matriz de Puntajes de las especies C. pneumnoniae y C.trachomatis

La Chlamydophila pneumoniae tiene 1112 CDS y la Chlamydia trachomatis solo 895 CDS. El alto número de ceros en la parte inicial del histograma indica la diferencia entre CDS entre cada una de ellas.

Otra forma de visualizar los datos es notando la posición de los puntos positivos en la matriz de puntuación.

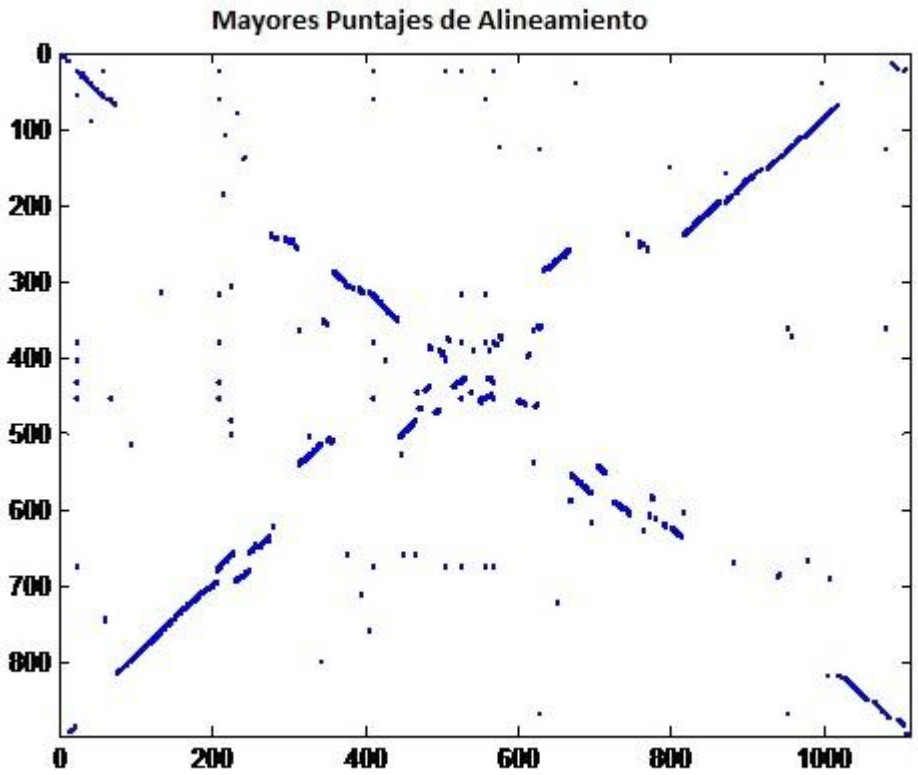

Ilustración 6. Mayores Puntajes de Alineamiento

El siguiente gráfico muestra la representación de los genes con el primer autovector seleccionado.

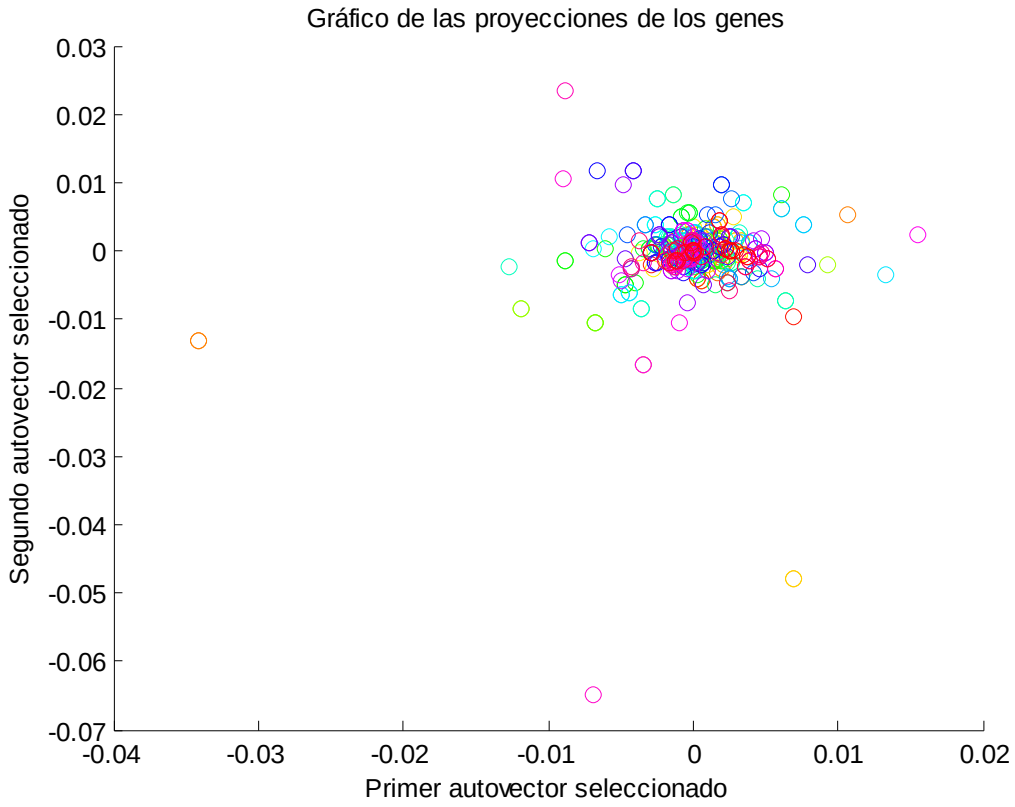

Ilustración 12. Proyecciones de los Genes

Los mejores cluster reportados por el algoritmo BBH y los peores constituyen las restricciones de tipo must-link y cannot-link, respectivamente, utilizadas en el semi-supervisado MPCKmeans.

Para observar los resultados de la aplicación del Co-agrupamiento Espectral con el algoritmo **MPCKMeans** véase **Anexo 7**.

3.8 Análisis de los resultados

Luego de llevar a cabo la experimentación se observa que en ambos casos junto con los ‘mejores’ cluster de puntuaciones mayores de 0.9 se incluyen otros genes con puntuación no muy buena. Es decir los grupos que reportan los ortólogos de alta puntuación contienen en su interior otros genes.

Mejores clusters (semi supervisado)

==========

Score: 0.950769
Chlamydia trachomatis Gene: 50S ribosomal protein L33
Chlamydophila pneumoniae Gen: 50S ribosomal protein L33

Score: 0.951087
Chlamydia trachomatis Gene: 30S ribosomal protein S21
Chlamydophila pneumoniae Gene: 30S ribosomal protein S21

Score: 0.930120
Chlamydia trachomatis Gene: 50S ribosomal protein L35
Chlamydophila pneumoniae Gene: 50S ribosomal protein L35

Score: 0.982993
Chlamydia trachomatis Gene: 50S ribosomal protein L36
Chlamydophila pneumoniae Gene: 50S ribosomal protein L36

Score: 0.932886
Chlamydia trachomatis Gene: 50S ribosomal protein L34
Chlamydophila pneumoniae Gene: 50S ribosomal protein L34

En los resultados anteriores se muestran los grupos reportados con una puntuación mayor que 0.9 que coinciden con los reportados por el BBH pero además poseen otros genes pertenecientes a estos grupos.

Por último se incluye todos los clusters obtenidos con el algoritmo semi-supervisado que contiene entre dos y ocho genes. En este caso se puede percibir que, en general, en cada cluster hay genes ‘parecidos’ según el valor del puntaje. Eso refleja la manera en que trabajan los métodos, es decir, en cada cluster se ubican objetos semejantes. Para observar los resultados de este experimento véase **Anexo 8**.

3.9 Conclusiones

Se seleccionó el algoritmo no supervisado KMeans y el algoritmo semi-supervisado MPCKMeans con los que se detectaron genes ortólogos en las especies Chlamydia trachomatis y Chlamydophila pneumoniae. Se describió como está almacenada la información de las secuencias, se detalló el método de Co-agrupamiento Espectral y se justificaron las herramientas utilizadas en la investigación. Se realizaron pruebas seleccionado varios autovectores y varias cantidades de clusters en cada caso. Los resultados que se obtienen son bastante similares y no fueron los esperados pues en los casos que se reporta puntuaciones altas los clusters poseen muchos genes en su composición.

Conclusiones Generales

1. Se analizaron herramientas computacionales para la detección de genes ortólogos las cuales aunque presentan buenos resultados son susceptibles de ser mejorados.
2. Se hizo una recopilación de información sobre la técnica semi-supervisada estudiando los diferentes enfoques que se proponen con vista a la implementación de algoritmos basados en la misma.
3. Se tomó como base para medir la calidad de los agrupamientos los resultados que reporta el algoritmo BBH.
4. Se realizó una transformación a las secuencias empleando el algoritmo Co-agrupamiento Espectral, con vista a obtener una representación vectorial de los genes.
5. Se empleó el algoritmo Co-agrupamiento Espectral junto con el KMean y el MPCKMeans para la detección de genes ortólogos en las especies Chlamydia trachomatis y la Chlamydophila pneumoniae.
6. Se compararon los resultados que reportaron los experimentos realizados.
7. La transformación de los datos asociada al método de Co-agrupamiento espectral no acentúa suficientemente las diferencias entre los genes.

Recomendaciones

1. Aplicar otros algoritmos de agrupamiento probabilístico con restricciones a las secuencias genómicas seleccionadas para comparar resultados con el MPCKMeans.

2. Aplicar algún método basado en grafos por ejemplo Propagación de Etiquetas que no necesitan la transformación de los datos pues trabajan directamente sobre las puntuaciones de alineamiento.

3. Analizar otras transformaciones para los datos y ampliar la experimentación.

4. Profundizar en el estudio de los datos que se poseen para aplicar otros algoritmos pertenecientes a otros enfoques de la técnica semi-supervisada.

5. Estudiar la posibilidad de conformar una base de casos seleccionando los rasgos que pudieran ser significativos para la detección de los genes ortólogos.

6. Aplicar los algoritmos semi-supervisados a otros organismos modelos como por ejemplo S. Cerevisiae y S.Pombe.

Referencias Bibliográficas

[1] Alberts, Biología molecular de la célula. Barcelona, 2005.

[2] E. M. P. D. Robertis, biología Celular y Molecular. Buenos Aires, 1998.

[3] (2011, Genética y homología Available: http://espanol.answers.yahoo.com/question/index?qid=20080509132803AAVPvOo

[4] M. E. Blay, "Herramientas Computacionales para la Comparación de Genomas," Universidad Central “Marta Abreu” de Las Villas., 2009.

[5] Sejnowski, Unsupervised Learning and Map Formation., 2009.

[6] (2010). Aprendizaje semi-supervisado utilizando algoritmos de agrupamiento.

[7] R. L. Tatusov, "The COG database: an updated version includes eukaryotes," BMC Bioinformatics, vol. 4, p. 14, 11 September 2003 2003.

[8] A. J. M. Feng Chen, Christian J. Stoeckert Jr and David S. Roos, "OrthoMCL-DB: querying a comprehensive multi-species collection of ortholog groups," Nucleic Acids Research, vol. 34, pp. D363-D368, 2006.

[9] I. T. Andrey Alexeyenko, Gang Liu and Erik L.L. Sonnhammer, "Automatic clustering of orthologs and inparalogs shared by multiple proteomes," Bioinformatics, vol. 22 pp. e9-e15, 2006.

[10] S. R. Lee Y, Pertea G, Cho J, Karamycheva S, Tsai J, Parvizi B, and A. V. Cheung F, White J et al., "Cross-referencing eukaryoticgenomes: TIGR Orthologous Gene Alignments (TOGA)," Genome Research, vol. 12, pp. 493-502, 2002.

[11] J. A. M. Christopher J. Penkett, Valerie Wood and Ju¨ rg Ba¨hler, "YOGY: a web-based, integrated database to retrieve protein orthologs and associated Gene Ontology terms," Nucleic Acids Research vol. 34, pp. W330-W334 2006.

[12] Deluca, et al., "Roundup: a multi-genome repository of orthologs and evolutionary distances," Bioinformatics, vol. 22, pp. 2044–2046, 2006.

[13] Logse. (Junio 1998, Biología Genética. Available: www.profes.net

[14] (30-10-2006, Glosario de Términos. Available: http://salud.glosario.net/terminos-medicos-de-enfermedades/cromosoma-2826.html

[15] S. Klug, Conceptos de Genética, octava ed.: Pearson Educación, 2006.

[16] Shamir and Frasconi, Artificial Intelligence and Heuristic Methods in Bioinformatics, 2003.

[17] M. Á. Cabrera. (2010, 02-2011). La Quimioinformática en el diseño in-silico de Fármacos. Available: http://molweb.cbq.uclv.edu.cu

[18] T. A. Vidal. (29/9/2009, 03-2011). Introducción a la Bioinformática II. Available: http://www.biologia.edu.ar/index.html

[19] T. Davis, Matrices y álgebra lineal, 2006.

[20] A. I. Odini, "Genómica Comparativa," ed, 2006.

[21] J. H. M. Díaz, "Propuesta de técnica de Inteligencia Artificial para la detección de anomalías en la red de datos de la Universidad de Cienfuegos.," Ingeniería en informática Investigativa, Universidad de Cienfuegos Carlos Rafael Rodriguez, 2010.

[22] L. A. F. Weidinger, Introducción a los alineamientos de secuencias, 2010.

[23] L. A. M. Soto, "Uso de Codones, Traducibilidad, Niveles de Expresión y Transferencia Horizontal: ¿Hemos Sobreinterpretado Nuestros Organismos Modelo?," Doctor en Cciencias Biomedicas, Centro de Ciencias Genomicas Programa de Genomica Computacional, Universidad Nacional Autonoma de Mexico, Mexico, junio de 2005.

[24] M. K. Kamvysselis, "Computational Comparative Genomics: Genes, Regulation, Evolution," Doctor of Philosophy in Computer Science, Massachusetts Institute of Technology, 2003.

[25] C. d. Paz. (2009, Analisis de secuencia de ADN. Available: //umber:sbs.man.ac.uk/dbbrowser/biactivity/

[26] D. M. M. G. Lorenzo, "Title," unpublished|.

[27] A. C. Berglund, et al., InParanoid 6: eukaryotic ortholog clusters with inparalogs. Linnaeus Centre for Bioinformatics, Uppsala University,, 2007.

[28] S. T. Ostlund G. (2009, InParanoid 7: new algorithms and tools for eukaryotic orthology analysis. Available: http://InParanoid.sbc.su.se

[29] Sourav Bandyopadhyay, 2 et al Roded Sharan,3,4 et al Trey Ideker1,2,4, "Systematic identification of functional orthologs
based on protein network comparison," p. 435, 2006.

[30] P. M. Groenen. (2006, Benchmarking ortholog identification methods using functional genomics data. Available: http://genomebiology.com/2006/7/4/R31

[31] P. Gogarten. (2007, 04 - 2011). BranchClust: un algoritmo filogenéticos para la selección de familias de genes Available: (http://creativecommons.org/licenses/by/2.0],

[32] T. Hulsen, et al. (2006, 02-2011). Benchmarking ortholog identification methods using functional genomics data. Available: http://genomebiology.com

[33] C. C. A. License. (2011, Neurosience Information Framework.

[34] O. University. (2010, 03-2011). HomoloGene. Available: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=homologene

[35] U. N. d. Colombia. (2010, 16-06-2011). Los Métodos WPGMA y UPGMA. Available: webmaster@unal.edu.co

[36] I. Uchiyama, Hierarchical clustering algorithm for comprehensive orthologous-domain classification in multiple genomes: Published online January 25, 2006, 2006.

[37] F. Roli, Semi-supervised Multiple Classifier Systems: Background and Research Directions. Italy, 2009.

[38] O. Chapelle, et al., Semi-Supervised Learning. The MIT Press Cambridge, Massachusetts London, England, 2006.

[39] A. B. Goldberg, "New Directions in SEMI-SUPERVISED LEARNING," Doctor of Philosophy, University of Wisconsin -Madison, 2010.

[40] X. Zhu, "Semi-Supervised Learning with Graphs," DOCTORAL THESIS, 2005.

[41] S. Basu, "Semi-supervised Clustering: Probabilistic Models, Algorithms and Experiments," Doctor of Philosophy, Presented to the Faculty of the Graduate School of The University of Texas at Austin, 2005.

[42] S. Basu, Semi-Supervisad Clustering by Seeding, July,2002.

[43] R. O. Duda, et al., Pattern Classification, segunda ed., 1997.

[44] R. XU, et al., "Clustering," 2009.

[45] I. Davidson, The Complexity of Non-Hierarchical Clustering With Instance and Cluster Level Constraints, 2007.

[46] E. Eaton, "Clustering with Propagated Constraints," Master of Science, University of Maryland, 2005.

[47] (2008, 16-02011). Que es MATLAB? Available: http//gemini.distrital.edu.co/index.php

[48] Weka, "Aprendizaje Automático," 3.4.12 ed, 2007.

[49] G. Xu, et al., Co-clustering Analysis of Weblogs Using Bipartite Spectral Projection Approach. Australia, 2010.

[50] B. K. kulis, et al., Semi-supervised Graph Clustering: A Kernel Approach, 2005.

[51] A. B. Korol, "Title," unpublished|.

[52] R. Cañedo. (2008, 05 - 2011). Single-link and complete-link clustering Available: nlp.stanford.edu

[53] H. Edelsbrunner. (2004, 04 - 2011). Average-Link Clustering. Available: www.dcs.gla.ac.uk

Bibliografía

[1](2010, 04 - 02 - 2011). Genómica comparativa - bioinformática. Available: www.molecularstation.com

[2]A. B. Goldberg, "New Directions in SEMI-SUPERVISED LEARNING," Doctor of Philosophy, University of Wisconsin -Madison, 2010.

[3]A. C. Berglund, et al., InParanoid 6: eukaryotic ortholog clusters with inparalogs. Linnaeus Centre for Bioinformatics, Uppsala University,, 2007.

[4]A. I. Odini, "Genómica Comparativa," ed, 2006.

[5]Alberts, Biología molecular de la célula. Barcelona, 2005.

[6]C. C. A. License. (2011, Neurosience Information Framework.)

[7]C. d. Paz. (2009, Analisis de secuencia de ADN. Available: //umber:sbs.man.ac.uk/dbbrowser/biactivity/

[8]Chapman and Hall, "Constrained Clustering Advances in Algorithms, Theory, and Applications," University of Minnesota.

[9]D. D. L. J. Aguilar, "Bioinformática y Salud Un nuevo paradigma de la sociedad."

[10]D. M. M. G. Lorenzo, "Title," unpublished|.

[11]D. P. Wall and T. DeLuca, Ortholog detection using the reciprocal smallest distance algorithm. Boston: The Computational Biology Initiative, Department of Systems Biology, Harvard Medical School.

[12]D. Sanchez, "Genética Humana," 2009.

[13]E. Eaton, "Clustering with Propagated Constraints," Master of Science, University of Maryland, 2005.

[14]E. M. P. D. Robertis, biología Celular y Molecular. Buenos Aires, 1998.

[15]F. Roli, Semi-supervised Multiple Classifier Systems: Background and Research Directions. Italy, 2009.

[16]G. Ostlund and T. Schmitt. (2009, InParanoid 7: new algorithms and tools for eukaryotic orthology analysis. Available: http://InParanoid.sbc.su.se

[17]G. Xu, et al., Co-clustering Analysis of Weblogs Using Bipartite Spectral Projection Approach. Australia, 2010.

[18]H. Edelsbrunner. (2004, 04 - 2011). Average -Link Clustering. Available: www.dcs.gla.ac.uk

[19]H. Zha, et al. (2008, 05-2011). Bipartite Graph Partitioning and Data Clustering.

[20]I. Davidson, The Complexity of Non-Hierarchical Clustering With Instance and Cluster Level Constraints, 2007.

[21]I. Uchiyama, Hierarchical clustering algorithm for comprehensive orthologous-domain classification in multiple genomes: Published online January 25, 2006, 2006.

[22]J. H. M. Díaz, "Propuesta de técnica de Inteligencia Artificial para la detección de anomalías en la red de datos de la Universidad de Cienfuegos.," Ingeniería en informática Investigativa, Universidad de Cienfuegos Carlos Rafael Rodriguez, 2010.

[23]J. L. Oliver Graph Clustering: A Kernel Approach, 2005.

[24]L. A. F. Weidinger, Introducción a los alineamientos de secuencias, 2010.

[25]L. A. M. Soto, "Uso de Codones, Traducibilidad, Niveles de Expresión y Transferencia Horizontal: ¿Hemos Sobreinterpretado Nuestros Organismos Modelo?," Doctor en Ciencias Biomedicas, Centro de Ciencias Genomicas Programa de Genomica Computacional, Universidad Nacional Autonoma de Mexico, Mexico, junio de 2005.

[26]Logse. (Junio 1998, Biología Genética. Available: www.profes.net

[27]M. Á. Cabrera. (2010, 02-2011). La Quimioinformática en el diseño in-silico de Fármacos. Available: http://molweb.cbq.uclv.edu.cu

[28]M. E. Blay, "Herramientas Computacionales para la Comparación de Genomas," Universidad Central “Marta Abreu” de Las Villas., 2009.

[29]M. K. Kamvysselis, "Computational Comparative Genomics: Genes, Regulation, Evolution," Doctor of Philosophy in Computer Science, Massachusetts Institute of Technology, 2003.

[30]O. Chapelle, et al., Semi-Supervised Learning. The MIT Press Cambridge, Massachusetts London, England, 2006.

[31]O. University. (2010, 03-2011). HomoloGene. Available: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=homologene

[32]P. Gogarten. (2007, 04 - 2011). BranchClust: un algoritmo filogenéticos para la selección de familias de genes Available: (http://creativecommons.org/licenses/by/2.0],

[33]P. L. e. I. Inza, "Clustering Particional y Jer´arquico," 2007.

[34]R. Cañedo. (2008, 05 - 2011). Single-link and complete-link clustering Available: nlp.stanford.edu

[35]R. O. Duda, et al., Pattern Classification, segunda ed., 1997.

[36]R. XU, et al., "Clustering," 2009.

[37]S. Bandyopadhyay, Systematic identification of functional orthologs based on protein network comparison. California, 2005.

[38]S. Basu, "Semi-supervised Clustering: Learning with Limited User Feedback," Doctor of Philosophy, University of Texas at Austin, November 2003.

[39]S. Basu, "Semi-supervised Clustering: Probabilistic Models, Algorithms and Experiments," Doctor of Philosophy, Presented to the Faculty of the Graduate School of The University of Texas at Austin, 2005.

[40]S. Basu, Semi-Supervisad Clustering by Seeding, July,2002.

[41]S. J. Russell and P. Norvig, INTELIGENCIA ARTIFICIAL UN ENFOQUE MODERNO, segunda edición ed. Madrid, 2004.

[42]Sejnowski, Unsupervised Learning and Map Formation,2009.

[43]Shamir and Frasconi, Artificial Intelligence and Heuristic Methods in Bioinformatics,2003.

[44]T. A. Vidal. (29/9/2009, 03-2011). Introducción a la Bioinformática II. Available: http://www.biologia.edu.ar/index.html

[45]T. Hulsen, et al. (2006, 02-2011). Benchmarking ortholog identification methods using functional genomics data. Available: http://genomebiology.com

[46]T. M. Mitchell, Generative and discriminative classifiers: Naive Bayes and Logistic Regression, 2006.

[47]X. Zhu,"Semi-Supervised Learning with Graphs," Doctoral Thesis, 2005.

[48]Z. FU, et al., "MSOAR: A High-Throughput Ortholog Assignment System Based on Genome Rearrangement," Journal of ComputacionalBiology, vol. 14, 2007.

Anexos

Anexo 1. Algoritmo EM

1Inicialización) $\theta^0, T, i = 0$

2 Hacer $i \leftarrow i + 1$

3 Paso E: calcular $Q(\theta; \theta^i)$

4 Paso M:$\theta^{i+1} \leftarrow \arg max\ Q(\theta; \theta^i)$

5 Hasta $Q(\theta^{i+1}; \theta^i) - Q(\theta^i; \theta^{i-1}) \leq T$

6 Devolver $\theta \leftarrow \theta^{i+1}$

7 Fin

Anexo 2.Resultados del algoritmo BBH

Score 0.982993
Chlamydia trachomatis Gene : 50S ribosomal protein L36
Chlamydophila pneumoniae Gene : 50S ribosomal protein L36
Score 0.981818
Chlamydia trachomatis Gene : 30S ribosomal protein S15
Chlamydophila pneumoniae Gene : 30S ribosomal protein S15
Score 0.975422
Chlamydia trachomatis Gene : Integration Host Factor Alpha
Chlamydophila pneumoniae Gene : integration host factor beta-subunit, putative
Score 0.971647

Chlamydia trachomatis Gene : 50S ribosomal protein L16
Chlamydophila pneumoniae Gene : 50S ribosomal protein L16
Score 0.970105
Chlamydia trachomatis Gene : 30S ribosomal protein S10
Chlamydophila pneumoniae Gene : 30S ribosomal protein S10
Score 0.969554
Chlamydia trachomatis Gene : Rod Shape Protein-Sugar Kinase
Chlamydophila pneumoniae Gene : cell shape-determining protein MreB
Score 0.953654
Chlamydia trachomatis Gene : hypothetical protein
Chlamydophila pneumoniae Gene : hypothetical protein
Score 0.953488
Chlamydia trachomatis Gene : translation initiation factor IF-1
Chlamydophila pneumoniae Gene : translation initiation factor IF-1
Score 0.951657
Chlamydia trachomatis Gene : elongation factor Tu
Chlamydophila pneumoniae Gene : elongation factor Tu
Score 0.951087
Chlamydia trachomatis Gene : 30S ribosomal protein S21
Chlamydophila pneumoniae Gene : 30S ribosomal protein S21
Score 0.950860
Chlamydia trachomatis Gene : 30S ribosomal protein S12

Chlamydophila pneumoniae Gene : 30S ribosomal protein S12
Score 0.950769
Chlamydia trachomatis Gene : 50S ribosomal protein L33
Chlamydophila pneumoniae Gene : 50S ribosomal protein L33
Score 0.946387
Chlamydia trachomatis Gene : ATP-dependent Clp protease proteolytic subunit
Chlamydophila pneumoniae Gene : ATP-dependent Clp protease proteolytic subunit
Score 0.945776
Chlamydia trachomatis Gene : 50S ribosomal protein L14
Chlamydophila pneumoniae Gene : 50S ribosomal protein L14
Score 0.944351
Chlamydia trachomatis Gene : 30S ribosomal protein S19
Chlamydophila pneumoniae Gene : 30S ribosomal protein S19
Score 0.942238
Chlamydia trachomatis Gene : 30S ribosomal protein S11
Chlamydophila pneumoniae Gene : 30S ribosomal protein S11
Score 0.940378
Chlamydia trachomatis Gene : type III secretion system ATPase
Chlamydophila pneumoniae Gene : type III secretion system ATPase
Score 0.940022
Chlamydia trachomatis Gene : elongation factor EF-2

Chlamydophila pneumoniae Gene : elongation factor EF-2
Score 0.938812
Chlamydia trachomatis Gene : chaperonin GroEL
Chlamydophila pneumoniae Gene : chaperonin GroEL
Score 0.938373
Chlamydia trachomatis Gene : elongation factor P
Chlamydophila pneumoniae Gene : elongation factor P
Score 0.932886
Chlamydia trachomatis Gene : 50S ribosomal protein L34
Chlamydophila pneumoniae Gene : 50S ribosomal protein L34
Score 0.930120
Chlamydia trachomatis Gene : 50S ribosomal protein L35
Chlamydophila pneumoniae Gene : 50S ribosomal protein L35
Score 0.929565
Chlamydia trachomatis Gene : Low Calcium Response D
Chlamydophila pneumoniae Gene : type III secretion inner membrane protein SctV
Score 0.928461
Chlamydia trachomatis Gene : DNA-directed RNA polymerase beta' subunit
Chlamydophila pneumoniae Gene : DNA-directed RNA polymerase beta' subunit
Score 0.923831
Chlamydia trachomatis Gene : DNA-directed RNA polymerase beta subunit

Chlamydophila pneumoniae Gene : DNA-directed RNA polymerase beta subunit
Score 0.920760
Chlamydia trachomatis Gene : ABC Transporter ATPase
Chlamydophila pneumoniae Gene : ABC transporter, ATP-binding protein
Score 0.918378
Chlamydia trachomatis Gene : GTP-binding protein LepA
Chlamydophila pneumoniae Gene : GTP-binding protein LepA
Score 0.916456
Chlamydia trachomatis Gene : transcription antitermination protein NusG
Chlamydophila pneumoniae Gene : transcription antitermination protein NusG
Score 0.916350
Chlamydia trachomatis Gene : hypothetical protein
Chlamydophila pneumoniae Gene : hypothetical protein
Score 0.916201
Chlamydia trachomatis Gene : 50S ribosomal protein L11
Chlamydophila pneumoniae Gene : 50S ribosomal protein L11
Score 0.914731
Chlamydia trachomatis Gene : transcription elongation factor NusA
Chlamydophila pneumoniae Gene : transcription elongation factor NusA
Score 0.913690
Chlamydia trachomatis Gene : hypothetical protein

Chlamydophila pneumoniae Gene : hypothetical protein

Anexo 3. Pseudocódigo del Algoritmo KMeans

Entrada: Conjunto de puntos de dato X=$\{x_i\}_{i-1}^{n}, x_i \epsilon R^d, k$ números de clusters

Salida: k particionamientos disjuntos $\{X_h\}_{h-1}^{k}$de X teniendo en cuenta la función objetivo.

Método:

1. Inicializar clusters: inicializar centroides $\{\mu_h^0\}_{h-1}^{k}$ son seleccionados aleatoriamente.

2. Repetir hasta la convergencia.

2a. assign_cluster: Asignar cada punto de dato x a el cluster h^*(por ejemplo conjunto $X_{h^*}^{(t+1)}$), para $h^*=argmin_h \|x-\mu_h^{(t)}\|^2$

2b. estimate_means: $\mu_h^{(t+1)} \leftarrow \frac{1}{X_h^{(t+1)}} \sum_{x \in X_h^{t+1}} X$

2c. $t \leftarrow (t+1)$

Anexo 4. Resultados de los Algoritmos Single Link y Complete Link (400 clusters)

Score 0.975422
Chlamydia trachomatis Gene : Integration Host Factor Alpha
Chlamydophila pneumoniae Gene : integration host factor beta-subunit, putative
Score 0.971647
Chlamydia trachomatis Gene : 50S ribosomal protein L16
Chlamydophila pneumoniae Gene : 50S ribosomal protein L16
Score 0.970105
Chlamydia trachomatis Gene : 30S ribosomal protein S10
Chlamydophila pneumoniae Gene : 30S ribosomal protein S10
Score 0.969554

Chlamydia trachomatis Gene : Rod Shape Protein-Sugar Kinase
Chlamydophila pneumoniae Gene : cell shape-determining protein MreB
Score 0.953654
Chlamydia trachomatis Gene : hypothetical protein
Chlamydophila pneumoniae Gene : hypothetical protein
Score 0.950860
Chlamydia trachomatis Gene : 30S ribosomal protein S12
Chlamydophila pneumoniae Gene : 30S ribosomal protein S12
Score 0.944351
Chlamydia trachomatis Gene : 30S ribosomal protein S19
Chlamydophila pneumoniae Gene : 30S ribosomal protein S19
Score 0.942238
Chlamydia trachomatis Gene : 30S ribosomal protein S11
Chlamydophila pneumoniae Gene : 30S ribosomal protein S11
Score 0.916456
Chlamydia trachomatis Gene : transcription antitermination protein NusG
Chlamydophila pneumoniae Gene : transcription antitermination protein NusG
Score 0.916350
Chlamydia trachomatis Gene : hypothetical protein
Chlamydophila pneumoniae Gene : hypothetical protein
Score 0.916201
Chlamydia trachomatis Gene : 50S ribosomal protein L11
Chlamydophila pneumoniae Gene : 50S ribosomal protein L11
Score 0.907998
Chlamydia trachomatis Gene : acetyl-CoA carboxylase alpha subunit
Chlamydophila pneumoniae Gene : acetyl-CoA carboxylase alpha subunit
Score 0.905000
Chlamydia trachomatis Gene : hypothetical protein

Chlamydophila pneumoniae Gene : hypothetical protein
Score 0.903394
Chlamydia trachomatis Gene : 30S ribosomal protein S13
Chlamydophila pneumoniae Gene : 30S ribosomal protein S13
Score 0.902073
Chlamydia trachomatis Gene : 50S ribosomal protein L2
Chlamydophila pneumoniae Gene : 50S ribosomal protein L2
15

Anexo 5. Resultados del Algoritmo Average Link (550 clusters)

Score 0.981818
Chlamydia trachomatis Gene : 30S ribosomal protein S15
Chlamydophila pneumoniae Gene : 30S ribosomal protein S15
Score 0.975422
Chlamydia trachomatis Gene : Integration Host Factor Alpha
Chlamydophila pneumoniae Gene : integration host factor beta-subunit, putative
Score 0.971647
Chlamydia trachomatis Gene : 50S ribosomal protein L16
Chlamydophila pneumoniae Gene : 50S ribosomal protein L16
Score 0.970105
Chlamydia trachomatis Gene : 30S ribosomal protein S10
Chlamydophila pneumoniae Gene : 30S ribosomal protein S10
Score 0.950860
Chlamydia trachomatis Gene : 30S ribosomal protein S12
Chlamydophila pneumoniae Gene : 30S ribosomal protein S12
Score 0.946387
Chlamydia trachomatis Gene : ATP-dependent Clp protease proteolytic subunit
Chlamydophila pneumoniae Gene : ATP-dependent Clp protease proteolytic subunit

Score 0.944351
Chlamydia trachomatis Gene : 30S ribosomal protein S19
Chlamydophila pneumoniae Gene : 30S ribosomal protein S19
Score 0.942238
Chlamydia trachomatis Gene : 30S ribosomal protein S11
Chlamydophila pneumoniae Gene : 30S ribosomal protein S11
Score 0.938373
Chlamydia trachomatis Gene : elongation factor P
Chlamydophila pneumoniae Gene : elongation factor P
Score 0.916456
Chlamydia trachomatis Gene : transcription antitermination protein NusG
Chlamydophila pneumoniae Gene : transcription antitermination protein NusG
Score 0.916350
Chlamydia trachomatis Gene : hypothetical protein
Chlamydophila pneumoniae Gene : hypothetical protein
Score 0.916201
Chlamydia trachomatis Gene : 50S ribosomal protein L11
Chlamydophila pneumoniae Gene : 50S ribosomal protein L11
Score 0.907998
Chlamydia trachomatis Gene : acetyl-CoA carboxylase alpha subunit
Chlamydophila pneumoniae Gene : acetyl-CoA carboxylase alpha subunit
Score 0.905000
Chlamydia trachomatis Gene : hypothetical protein
Chlamydophila pneumoniae Gene : hypothetical protein
Score 0.903394
Chlamydia trachomatis Gene : 30S ribosomal protein S13
Chlamydophila pneumoniae Gene : 30S ribosomal protein S13

Score 0.903226
Chlamydia trachomatis Gene : Yop proteins translocation protein S
Chlamydophila pneumoniae Gene : type III secretion inner membrane protein SctS
16

Anexo 6. Resultados del Algoritmo KMeans (500 clusters)

Score 0.981818
Chlamydia trachomatis Gene : 30S ribosomal protein S15
Chlamydophila pneumoniae Gene : 30S ribosomal protein S15
Score 0.975422
Chlamydia trachomatis Gene : Integration Host Factor Alpha
Chlamydophila pneumoniae Gene : integration host factor beta-subunit, putative
Score 0.971647
Chlamydia trachomatis Gene : 50S ribosomal protein L16
Chlamydophila pneumoniae Gene : 50S ribosomal protein L16
Score 0.953488
Chlamydia trachomatis Gene : translation initiation factor IF-1
Chlamydophila pneumoniae Gene : translation initiation factor IF-1
Score 0.946387
Chlamydia trachomatis Gene : ATP-dependent Clp protease proteolytic subunit
Chlamydophila pneumoniae Gene : ATP-dependent Clp protease proteolytic subunit
Score 0.944351
Chlamydia trachomatis Gene : 30S ribosomal protein S19
Chlamydophila pneumoniae Gene : 30S ribosomal protein S19
Score 0.942238
Chlamydia trachomatis Gene : 30S ribosomal protein S11

Chlamydophila pneumoniae Gene : 30S ribosomal protein S11
Score 0.930120
Chlamydia trachomatis Gene : 50S ribosomal protein L35
Chlamydophila pneumoniae Gene : 50S ribosomal protein L35
Score 0.929565
Chlamydia trachomatis Gene : Low Calcium Response D
Chlamydophila pneumoniae Gene : type III secretion inner membrane protein SctV
Score 0.916456
Chlamydia trachomatis Gene : transcription antitermination protein NusG
Chlamydophila pneumoniae Gene : transcription antitermination protein NusG
Score 0.914731
Chlamydia trachomatis Gene : transcription elongation factor NusA
Chlamydophila pneumoniae Gene : transcription elongation factor NusA
Score 0.913690
Chlamydia trachomatis Gene : hypothetical protein
Chlamydophila pneumoniae Gene : hypothetical protein
Score 0.930120
Chlamydia trachomatis Gene : 50S ribosomal protein L35
Chlamydophila pneumoniae Gene : 50S ribosomal protein L35
Score 0.929565
Chlamydia trachomatis Gene : Low Calcium Response D
Chlamydophila pneumoniae Gene : type III secretion inner membrane protein SctV
Score 0.916456
Chlamydia trachomatis Gene : transcription antitermination protein NusG
Chlamydophila pneumoniae Gene : transcription antitermination protein NusG
Score 0.914731

Chlamydia trachomatis Gene : transcription elongation factor NusA
Chlamydophila pneumoniae Gene : transcription elongation factor NusA
Score 0.913690
Chlamydia trachomatis Gene : hypothetical protein
Chlamydophila pneumoniae Gene : hypothetical protein
Score 0.900534
Chlamydia trachomatis Gene : acetyl-CoA carboxylase
Chlamydophila pneumoniae Gene : acetyl-CoA carboxylase
18

Anexo 7. Resultados de la aplicación del Co-agrupamiento Espectral con el algoritmo MPCKMeans

A continuación se incluyen los 100 clusters generados por el MPCKMeans con la cantidad de genes en cada uno.

100

Total de genes en el cluster 1 = 29

Total de genes en el cluster 2 = 42

Total de genes en el cluster 3 = 18

Total de genes en el cluster 4 = 20

Total de genes en el cluster 5 = 12

Total de genes en el cluster 6 = 31

Total de genes en el cluster 7 = 2

Score 0.009828

Total de genes en el cluster 8 = 24

Total de genes en el cluster 9 = 16

Total de genes en el cluster 10 = 6

Total de genes en el cluster 11 = 38

Total de genes en el cluster 12 = 6

Total de genes en el cluster 13 = 8

Total de genes en el cluster 14 = 15

Total de genes en el cluster 15 = 22

Total de genes en el cluster 16 = 13

Total de genes en el cluster 17 = 74

Total de genes en el cluster 18 = 8

Total de genes en el cluster 19 = 2

Score 0.608479

Total de genes en el cluster 20 = 21

Total de genes en el cluster 21 = 12

Total de genes en el cluster 22 = 19

Total de genes en el cluster 23 = 2

Score 0.520216

Total de genes en el cluster 24 = 8

Total de genes en el cluster 25 = 3

Total de genes en el cluster 26 = 8

Total de genes en el cluster 27 = 12

Total de genes en el cluster 28 = 6

Total de genes en el cluster 29 = 4

Total de genes en el cluster 30 = 4

Total de genes en el cluster 31 = 10

Total de genes en el cluster 32 = 5

Total de genes en el cluster 33 = 4

Total de genes en el cluster 34 = 2

Score 0.710247

Total de genes en el cluster 35 = 14

Total de genes en el cluster 36 = 873

Total de genes en el cluster 37 = 3

Total de genes en el cluster 38 = 0

Total de genes en el cluster 39 = 0

Total de genes en el cluster 40 = 0

Total de genes en el cluster 41 = 0

Total de genes en el cluster 42 = 0

Total de genes en el cluster 43 = 4

Total de genes en el cluster 44 = 0

Total de genes en el cluster 45 = 0

Total de genes en el cluster 46 = 4

Total de genes en el cluster 47 = 14

Total de genes en el cluster 48 = 2

Score 0.664298

Total de genes en el cluster 49 = 40

Total de genes en el cluster 50 = 14

Total de genes en el cluster 51 = 0

Total de genes en el cluster 52 = 2

Score 0.008547

Total de genes en el cluster 53 = 28

Total de genes en el cluster 54 = 33

Total de genes en el cluster 55 = 20

Total de genes en el cluster 56 = 15

Total de genes en el cluster 57 = 2

Score 0.019560

Total de genes en el cluster 58 = 2

Score 0.027491

Total de genes en el cluster 59 = 0

Total de genes en el cluster 60 = 21

Total de genes en el cluster 61 = 8

Total de genes en el cluster 62 = 0

Total de genes en el cluster 63 = 18

Total de genes en el cluster 64 = 16

Total de genes en el cluster 65 = 0

Total de genes en el cluster 66 = 0

Total de genes en el cluster 67 = 0

Total de genes en el cluster 68 = 0

Total de genes en el cluster 69 = 27

Total de genes en el cluster 70 = 0

Total de genes en el cluster 71 = 0

Total de genes en el cluster 72 = 10

Total de genes en el cluster 73 = 0

Total de genes en el cluster 74 = 33

Total de genes en el cluster 75 = 10

Total de genes en el cluster 76 = 2

Score 0.031886

Total de genes en el cluster 77 = 8

Total de genes en el cluster 78 = 8

Total de genes en el cluster 79 = 0

Total de genes en el cluster 80 = 2

Score 0.046154

Total de genes en el cluster 81 = 8

Total de genes en el cluster 82 = 12

Total de genes en el cluster 83 = 0

Total de genes en el cluster 84 = 2

Score 0.003399

Total de genes en el cluster 85 = 8

Total de genes en el cluster 86 = 6

Total de genes en el cluster 87 = 0

Total de genes en el cluster 88 = 8

Total de genes en el cluster 89 = 0

Total de genes en el cluster 90 = 0

Total de genes en el cluster 91 = 0

Total de genes en el cluster 92 = 0

Total de genes en el cluster 93 = 0

Total de genes en el cluster 94 = 4

Total de genes en el cluster 95 = 13

Total de genes en el cluster 96 = 0

Total de genes en el cluster 97 = 0

Total de genes en el cluster 98 = 8

Total de genes en el cluster 99 = 0

Total de genes en el cluster 100 = 4

Anexo 8. Resultados del MPCKMeans con otros parámetros

Cluster(semi supervisado): 7

Total de genes en el cluster: 2

==========

Score: 0.009828

Chlamydia trachomatis Gene: hypothetical protein

Chlamydophila pneumoniae Gene: hypothetical protein

Cluster(semi supervisado): 10

Total de genes en el cluster: 6

==========

Score: 0.568771

Chlamydia trachomatis Gene: rRNA Methylase (SpoU family)

Chlamydophila pneumoniae Gene: spoU rRNA methylase family protein

Score: 0.146829

Chlamydia trachomatis Gene: hypothetical protein

Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.465711

Chlamydia trachomatis Gene: lipid A biosynthesis lauroyl acyltransferase

Chlamydophila pneumoniae Gene: lipid A biosynthesis lauroyl acyltransferase

Cluster(semi supervisado): 12

Total de genes en el cluster: 6

==========

Score: 0.419003

Chlamydia trachomatis Gene: Ppheopstpihdoel ipase D superfamily [uncleavable]

Chlamydophila pneumoniae Gene: phospholipase D family protein

Score: 0.092221

Chlamydia trachomatis Gene: hypothetical protein

Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.453397

Chlamydia trachomatis Gene: hypothetical protein

Chlamydophila pneumoniae Gene: hypothetical protein

Cluster(semi supervisado): 13

Total de genes en el cluster: 8

==========

Score: 0.678234

Chlamydia trachomatis Gene: transglycolase/transpeptidase

Chlamydophila pneumoniae Gene: penicillin-binding protein

Score: 0.716060

Chlamydia trachomatis Gene: isoleucyl-tRNA synthetase

Chlamydophila pneumoniae Gene: isoleucyl-tRNA synthetase

Score: 0.738182

Chlamydia trachomatis Gene: valyl-tRNA synthetase

Chlamydophila pneumoniae Gene: valyl-tRNA synthetase

Score: 0.629156
Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: hypothetical protein

Cluster(semi supervisado): 18
Total de genes en el cluster: 8
==========

Score: 0.507831
Chlamydia trachomatis Gene: apolipoprotein N-acyltransferase
Chlamydophila pneumoniae Gene: apolipoprotein N-acyltransferase

Score: 0.807461
Chlamydia trachomatis Gene: glucose-inhibited division protein A
Chlamydophila pneumoniae Gene: glucose-inhibited division protein A

Score: 0.697320
Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.389717
Chlamydia trachomatis Gene: 4-hydroxybenzoate octaprenyltransferase
Chlamydophila pneumoniae Gene: 4-hydroxybenzoate octaprenyltransferase

Cluster(semi supervisado): 19
Total de genes en el cluster: 2
==========

Score: 0.608479

Chlamydia trachomatis Gene: tprraontsecirni p t cleavage factor / unknown do main fusion
Chlamydophila pneumoniae Gene: tprraontsecirni p t cleavage factor / unknown do main fusion

Cluster(semi supervisado): 23
Total de genes en el cluster: 2
==========
Score: 0.520216
Chlamydia trachomatis Gene: DNA mismatch repair protein
Chlamydophila pneumoniae Gene: DNA mismatch repair protein

Cluster(semi supervisado): 24
Total de genes en el cluster: 8
==========
Score: 0.631339
Chlamydia trachomatis Gene: GutQ/KpsF Family Sugar-P Isomerase
Chlamydophila pneumoniae Gene: carbohydrate isomerase, KpsF/GutQ family

Score: 0.583104
Chlamydia trachomatis Gene : 4s-yhnytdhraosxey - 3 – methylbut-2-en-1
Chlamydophila pneumoniae Gene: 4s-yhnytdhraosxey - 3 - methylbut-2-en-1Score: 0.644184
Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.604567
Chlamydia trachomatis Gene: 3-deoxy-manno-octulosonate cytidylyltransferase
Chlamydophila pneumoniae Gene: 3-deoxy-manno-octulosonate cytidylyltransferase

Cluster(semi supervisado): 25

Total de genes en el cluster: 3

==========

Score: 0.214953

Chlamydia trachomatis Gene: hypothetical protein

Chlamydophila pneumoniae Gene: hypothetical protein

Cluster(semi supervisado): 26

Total de genes en el cluster: 8

==========

Score: 0.529926

Chlamydia trachomatis Gene: Protease

Chlamydophila pneumoniae Gene: protease IV, putative

Score: 0.543126

Chlamydia trachomatis Gene: CHLPS Euo Protein

Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.754114

Chlamydia trachomatis Gene: 3-oxoacyl-(acyl carrier protein) synthase

Chlamydophila pneumoniae Gene: 3-oxoacyl-(acyl carrier protein) synthase

Score: 0.566385

Chlamydia trachomatis Gene: multidomain protein family

Chlamydophila pneumoniae Gene: hypothetical protein

Cluster(semi supervisado): 28

Total de genes en el cluster: 6

==========

Score: 0.928461

Chlamydia trachomatis Gene: DNA-directed RNA polymerase beta' subunit
Chlamydophila pneumoniae Gene: DNA-directed RNA polymerase beta' subunit

Score: 0.529679
Chlamydia trachomatis Gene: Thio:disulfide Interchange Protein
Chlamydophila pneumoniae Gene: thiol:disulfide interchange protein DsbD

Score: 0.632378
Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: hypothetical protein

Cluster(semi supervisado): 29
Total de genes en el cluster: 4
==========
Score: 0.586873
Chlamydia trachomatis Gene: translocation protein TolB precursor
Chlamydophila pneumoniae Gene: translocation protein TolB precursor

Score: 0.527183
Chlamydia trachomatis Gene: Dihydropteroate Synthase
Chlamydophila pneumoniae Gene: 2p-yarmoipnhoo-s4p-hhoykdirnoaxsye-/6d-ihhyyddrrooxpytmeertohaytled ishyyndtrhoapstee r i n

Cluster(semi supervisado): 30
Total de genes en el cluster: 4
==========
Score: 0.703564
Chlamydia trachomatis Gene: methionyl-tRNA synthetase
Chlamydophila pneumoniae Gene: methionyl-tRNA synthetase

Score: 0.698050
Chlamydia trachomatis Gene: PTS IIA Protein + HTH DNA-Binding Domain
Chlamydophila pneumoniae Gene: PTS system, IIA component

Cluster(semi supervisado): 32
Total de genes en el cluster: 5
==========

Score: 0.007802
Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.522151
Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.875824
Chlamydia trachomatis Gene: RNA polymerase sigma factor
Chlamydophila pneumoniae Gene: RNA polymerase sigma factor

Cluster(semi supervisado): 33
Total de genes en el cluster: 4
==========

Score: 0.737984
Chlamydia trachomatis Gene: glutamyl-tRNA amidotransferase subunit A
Chlamydophila pneumoniae Gene: glutamyl-tRNA amidotransferase subunit A

Score: 0.724247
Chlamydia trachomatis Gene: predicted oxidoreductase
Chlamydophila pneumoniae Gene: hypothetical protein

Cluster(semi supervisado): 34

Total de genes en el cluster: 2

==========

Score: 0.710247

Chlamydia trachomatis Gene: DNA topoisomerase I/SWI domain fusion protein

Chlamydophila pneumoniae Gene: DNA topoisomerase I/SWI domain fusion protein

Cluster(semi supervisado): 37

Total de genes en el cluster: 3

==========

Score: 0.656360

Chlamydia trachomatis Gene: C3D-Pp-hdoisapchyaltgildyycletrroaln-s-fgelryacseer ol-3-phosphate

Chlamydophila pneumoniae Gene: C3D-Pp-hdoisapchyaltgildyycletrroaln-s-fgelryacseer oplr-o3t-epihno,s pphuattaetive

Score: 0.844624

Chlamydia trachomatis Gene: hypothetical protein

Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.659425

Chlamydia trachomatis Gene: 3-ketoacyl-(acyl-carrier-protein) reductase

Chlamydophila pneumoniae Gene: 3-ketoacyl-(acyl-carrier-protein) reductase

Cluster(semi supervisado): 43

Total de genes en el cluster: 4

==========

Score: 0.258393

Chlamydia trachomatis Gene: ABC Transporter Membrane Protein

Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.527761

Chlamydia trachomatis Gene: N-Acetylmuramoyl Alanine Amidase

Chlamydophila pneumoniae Gene: N-acetylmuramoyl-L-alanine amidase, putative

Cluster(semi supervisado): 46

Total de genes en el cluster: 4

==========

Score: 0.038827

Chlamydia trachomatis Gene: Inclusion Membrane Protein C

Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.015531

Chlamydia trachomatis Gene: Inclusion Membrane Protein C

Chlamydophila pneumoniae Gene: inclusion membrane protein B

Score: 0.013805

Chlamydia trachomatis Gen : Inclusion Membrane Protein C

Chlamydophila pneumoniae Gene: hypothetical protein

Cluster(semi supervisado): 48

Total de genes en el cluster: 2

==========

Score: 0.664298

Chlamydia trachomatis Gene: hypothetical protein

Chlamydophila pneumoniae Gene: hypothetical protein

Cluster(semi supervisado): 52

Total de genes en el cluster: 2

==========

Score: 0.008547

Chlamydia trachomatis Gene: hypothetical protein

Chlamydophila pneumoniae Gene: hypothetical protein

Cluster(semi supervisado): 57

Total de genes en el cluster: 2

==========

Score: 0.019560

Chlamydia trachomatis Gene: hypothetical protein

Chlamydophila pneumoniae Gene: hypothetical protein

Cluster(semi supervisado): 58

Total de genes en el cluster: 2

==========

Score: 0.027491

Chlamydia trachomatis Gene: hypothetical protein

Chlamydophila pneumoniae Gene: hypothetical protein

Cluster(semi supervisado): 61

Total de genes en el cluster: 8

==========

Score: 0.093880

Chlamydia trachomatis Gene: hypothetical protein

Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.779825

Chlamydia trachomatis Gene: glyceraldehyde-3-phosphate dehydrogenase

Chlamydophila pneumoniae Gene: glyceraldehyde-3-phosphate dehydrogenase

Score: 0.617002

Chlamydia trachomatis Gene: aspartate-semialdehyde dehydrogenase

Chlamydophila pneumoniae Gene: aspartate-semialdehyde dehydrogenase

Score: 0.720682
Chlamydia trachomatis Gene: predicted D-Amino Acid Dehydrogenase
Chlamydophila pneumoniae Gene: oxidoreductase, DadA family

Cluster(semi supervisado): 76
Total de genes en el cluster: 2
==========

Score: 0.031886
Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: hypothetical protein

Cluster(semi supervisado): 77
Total de genes en el cluster: 8
==========

Score: 0.415702
Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.868799
Chlamydia trachomatis Gene: 30S ribosomal protein S1
Chlamydophila pneumoniae Gene: 30S ribosomal protein S1

Score: 0.311394
Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.731681
Chlamydia trachomatis Gene: dneuocxlyeuortiiddionhey d5r'o-ltarsiep hosphate
Chlamydophila pneumoniae Gene: dneuocxlyeuortiiddionhey d5r'o-ltarsiep hosphate

Cluster(semi supervisado): 78

Total de genes en el cluster: 8

==========

Score: 0.457831

Chlamydia trachomatis Gene: hypothetical protein

Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.313949

Chlamydia trachomatis Gene: tetraacyldisaccharide 4'-kinase

Chlamydophila pneumoniae Gene: tetraacyldisaccharide 4'-kinase

Score: 0.729891

Chlamydia trachomatis Gene: predicted metal dependent hydrolase

Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.522154

Chlamydia trachomatis Gene: DNA polymerase III subunits gamma and tau

Chlamydophila pneumoniae Gene: DNA polymerase III subunits gamma and tau

Cluster(semi supervisado): 80

Total de genes en el cluster: 2

==========

Score: 0.046154

Chlamydia trachomatis Gene: hypothetical protein

Chlamydophila pneumoniae Gene: hypothetical protein

Cluster(semi supervisado): 81

Total de genes en el cluster: 8

==========

Score: 0.635580

Chlamydia trachomatis Gene: Oligopeptide Permease
Chlamydophila pneumoniae Gene: ppeupttaitdiev eABC transporter,permeaseprotein,

Score: 0.492908
Chlamydia trachomatis Gene: HAD superfamily hydrolase/phosphatase
Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.590791
Chlamydia trachomatis Gene: Thioredoxin Reductase
Chlamydophila pneumoniae Gene: thioredoxin reductase

Score: 0.437388
Chlamydia trachomatis Gene: Dihydrofolate Reductase
Chlamydophila pneumoniae Gene: dihydrofolate reductase

Cluster(semi supervisado): 84
Total de genes en el cluster: 2
==========
Score: 0.003399
Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: hypothetical protein

Cluster(semi supervisado): 85
Total de genes en el cluster: 8
==========
Score: 0.108796
Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.429299

Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.097847
Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.373400
Chlamydia trachomatis Gene: triosephosphate isomerase
Chlamydophila pneumoniae Gene: triosephosphate isomerase

Cluster(semi supervisado): 86
Total de genes en el cluster: 6
==========
Score: 0.028037
Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.014152
Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.413613
Chlamydia trachomatis Gene: dihydrodipicolinate synthase
Chlamydophila pneumoniae Gene: dihydrodipicolinate synthase

Cluster(semi supervisado): 88
Total de genes en el cluster: 8
==========
Score: 0.534634

Chlamydia trachomatis Gene: Nsau(b+u)n-ittr aCn slocating NADH-quinone reductase
Chlamydophila pneumoniae Gene: Nsau(b+u)n-ittr aCn slocating NADH-quinone reductase

Score: 0.493464
Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: thiamine biosynthesis lipoprotein

Score: 0.383306
Chlamydia trachomatis Gene: acid phosphatase
Chlamydophila pneumoniae Gene: acid phosphatase

Score: 0.307532
Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: hypothetical protein

Cluster(semi supervisado): 94
Total de genes en el cluster: 4
==========
Score: 0.001735
Chlamydia trachomatis Gene: Membrane Thiol Protease (predicted)
Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.014706
Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: hypothetical protein

Cluster(semi supervisado): 98
Total de genes en el cluster: 8
==========

Score: 0.708222
Chlamydia trachomatis Gene: Glucose-1-P Adenyltransferase
Chlamydophila pneumoniae Gene: glucose-1-phosphate adenylyltransferase

Score: 0.228627
Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.628595
Chlamydia trachomatis Gene: 1-deoxy-D-xylulose 5-phosphate reductoisomerase
Chlamydophila pneumoniae Gene: 1-deoxy-D-xylulose 5-phosphate reductoisomerase

Score: 0.477764
Chlamydia trachomatis Gene: ribose-5-phosphate isomerase A
Chlamydophila pneumoniae Gene: ribose-5-phosphate isomerase A

Cluster(semi supervisado): 100
Total de genes en el cluster: 4
==========
Score: 0.301574
Chlamydia trachomatis Gene: hypothetical protein
Chlamydophila pneumoniae Gene: hypothetical protein

Score: 0.391975
Chlamydia trachomatis Gene: prreepdeiactt efda mpiollyy saccharide hydrolase-invasin
Chlamydophila pneumoniae Gene: NLP/P60 family protein